PREMIER REFERENCE SOURCE

Railway Safety, Reliability and Security

Technologies and Systems Engineering



Francesco Flammini

Railway Safety, Reliability, and Security:

Technologies and Systems Engineering

Francesco Flammini
Ansaldo STS, Italy

Information Science
REFERENCE

Managing Director: Lindsay Johnston
Senior Editorial Director: Heather A. Probst
Book Production Manager: Sean Woznicki
Development Manager: Joel Gamon
Development Editor: Myla Harty
Acquisitions Editor: Erika Gallagher
Typesetter: Nicole Sparano
Cover Design: Nick Newcomer, Lisandro Gonzalez

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Copyright © 2012 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Railway safety, reliability, and security: technologies and systems engineering / Francesco Flammini, editor.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-4666-1643-1 (hardcover) -- ISBN 978-1-4666-1644-8 (ebook) -- ISBN 978-1-4666-1645-5 (print & perpetual access) 1. Railroads--Safety measures. 2. Railroads--Security measures. I. Flammini, Francesco.
TF610.R37 2012
625.10028’9--dc23

2011050505

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Editorial Advisory Board

Marina Aguado, *University of the Basque Country UPV/EHU, Spain*
Alfredo Benso, *Politecnico di Torino, Italy*
Jens Braband, *Siemens AG, Germany*
Calin Ciufudean, *Stefan cel Mare University, Romania*
Simon Collart-Dutilleul, *LAGIS-INRETS, France*
Stefano Di Carlo, *Politecnico di Torino, Italy*
Christian Dufour, *Opal-RT Technologies, Canada*
Lars Ebrecht, *German Aerospace Center (DLR), Germany*
Alessandro Fantechi, *DSI - Universita' di Firenze, Itlay*
Johannes Feuser, *University of Bremen, Germany*
Matthias Guedemann, *University of Magdeburg, Germany*
Mark Hartong, *Federal Railroad Administration, USA*
Luigi Iannelli, *University of Sannio, Italy*
Ronald Lindsey, *Communication Architecture, USA*
Stefano Marrone, *Second University of Naples, Italy*
Jaizki Mendizabal, *CEIT and Tecnun, University of Navarra, Spain*
Jon Mendizabal-Samper, *CEIT and Tecnun, University of Navarra, Spain*
Michael Meyer Zu Hörste, *German Aerospace Center, Germany*
Frank Ortmeier, *University of Magdeburg, Germany*
Jan Peleska, *University of Bremen, Germany*
Margarita Peltekova, *University of Trier, Germany*
Dorian Petit, *LAMIH- University of Valenciennes and Hainaut- Cambresis, France*
Vincent Poirriez, *LAMIH- University of Valenciennes and Hainaut- Cambresis, France*
Luca Pugi, *University of Florence, Italy*
Wolfgang Reif, *University of Augsburg, Germany*
Clive Roberts, *University of Birmingham, UK*
Alessandro Savino, *Politecnico di Torino, Italy*
Jörn Guy Süß, *The University of Queensland, Australia*
Francesco Vasca, *University of Sannio, Italy*
Valeria Vittorini, *Università "Federico II" di Napoli, Italy*
Duminda Wijesekera, *George Mason University, USA*
Kirsten Winter, *George Mason University, USA*
Christian Wullems, *Cooperative Research Centre for Rail Innovation, Australia*
Wei Zheng, *Beijing Jiaotong University, China*
Armin Zimmermann, *Technische Universitaet Ilmenau, Germany*

Table of Contents

Foreword by Odd Nordland xvii

Foreword by Stephan Jubin xix

Preface xxi

Section 1
Regulations and Certification Standards

Chapter 1
U.S. Regulatory Requirements for Positive Train Control Systems 1
Mark Hartong, Federal Railroad Administration, USA
Duminda Wijesekera, George Mason University, USA

Chapter 2
The Model-Driven openETCS Paradigm for Secure, Safe and Certifiable Train Control Systems 22
Jan Peleska, University of Bremen, Germany
Johannes Feuser, University of Bremen, Germany
Anne E. Haxthausen, Technical University of Denmark, Denmark

Section 2
Hazard Analysis and Model-Based Evaluation

Chapter 3
Semi-Quantitative Risk Assessment of Technical Systems on European Railways 54
Jens Braband, Siemens AG, Germany

Chapter 4
The ForMoSA Approach to Qualitative and Quantitative Model-Based Safety Analysis 65
Axel Habermaier, Universität Augsburg, Institut für Informatik, Germany
Matthias Güdemann, Otto-von-Guericke University of Magdeburg, Germany
Frank Ortmeier, Otto-von-Guericke University of Magdeburg, Germany
Wolfgang Reif, Universität Augsburg, Institut für Informatik, Germany
Gerhard Schellhorn, Universität Augsburg, Institut für Informatik, Germany

Section 3
Verification and Validation

Chapter 5
Verification and Validation of Interoperability 116
Lars Ebrecht, DLR (German Aerospace Center), Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, DLR (German Aerospace Center), Institute of Transportation Systems, Germany

Chapter 6
Fault Injection for On-Board ERTMS/ETCS Safety Assessment 128
Almir Villaro Arriola, CEIT and Tecnun (University of Navarra), Spain
Jon Mendizabal Samper, CEIT and Tecnun (University of Navarra), Spain
Juan Meléndez Lagunilla, CEIT and Tecnun (University of Navarra), Spain

Chapter 7
Impact of Electromagnetic Environment on Reliability Assessment for Railway Signalling Systems 151
Iñigo Adin, CEIT and Tecnun (University of Navarra), Spain
Jaizki Mendizabal, CEIT and Tecnun (University of Navarra), Spain
Jon del Portillo, CEIT and Tecnun (University of Navarra), Spain

Section 4
Automation in Development and Testing

Chapter 8
Μίνθα: A Framework for Auto-Programming and Testing of Railway Controllers for Varying Clients 175
Jörn Guy Süß, University of Queensland, Australia
Neil Robinson, RGB Assurance, Australia
David Carrington, University of Queensland, Australia
Paul Strooper, University of Queensland, Australia

Chapter 9
Software-Based Self-Test for Reliable Applications in Railway Systems 198
Alfredo Benso, Politecnico di Torino, Italy
Stefano Di Carlo, Politecnico di Torino, Italy
Alessandro Savino, Politecnico di Torino, Italy

Chapter 10
Real-Time Hardware-in-the-Loop in Railway: Simulations for Testing Control Software of Electromechanical Train Components 221
Silvio Baccari, University of Sannio, Italy
Giulio Cammeo, AnsaldoBreda, Italy
Christian Dufour, Opal-RT Technologies, Canada
Luigi Iannelli, University of Sannio, Italy
Vincenzo Mungiguerra, AnsaldoBreda, Italy
Mario Porzio, AnsaldoBreda, Italy
Gabriella Reale, University of Sannio, Italy
Francesco Vasca, University of Sannio, Italy

Chapter 11
Hardware-In-the-Loop Testing of On-Board Subsystems: Some Case Studies and Applications 249
Luca Pugi, University of Florence, Italy
Benedetto Allotta, University of Florence, Italy

Section 5
Formal Methods

Chapter 12
The Role of Formal Methods in Software Development for Railway Applications 282
Alessandro Fantechi, Università degli Studi di Firenze, Italy

Chapter 13
Symbolic Model Checking for Interlocking Systems 298
Kirsten Winter, The University of Queensland, Australia

Section 6
Human Factors

Chapter 14
Designing Usable Interactive Systems within the Railway Domain: A Human Factors Approach 317
Nina Jellentrup, German Aerospace Center, Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, German Aerospace Center, Institute of Transportation Systems, Germany

Chapter 15
Integration of Human Factors to Safety Assessments by Human Barrier Interaction 327
Markus Talg, German Aerospace Center, Institute of Transportation Systems, Germany
Malte Hammerl, German Aerospace Center, Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, German Aerospace Center, Institute of Transportation Systems, Germany

Section 7
Security, Monitoring and Surveillance

Chapter 16
Advanced Techniques for Monitoring the Condition of Mission-Critical Railway Equipment 341
Clive Roberts, University of Birmingham, UK
Joe Silmon, University of Birmingham, UK

Chapter 17
Security of Railway Infrastructures 355
A. Di Febbraro, University of Genoa, Italy
F. Papa, University of Genoa, Italy
N. Sacco, University of Genoa, Italy

Section 8
Experiences and Case-Studies

Chapter 18
ETCS Developing and Operation: Italian Experience 381
Raffaele Malangone, RFI, Italy
Fabio Senesi, ANSF, Italy

Chapter 19
Adoption of Low-Cost Rail Level Crossing Warning Devices: An Australian Case Study 399
Christian Wullems, Cooperative Research Centre for Rail Innovation, & Centre for Accident Research and Road Safety – Queensland (CARRS-Q), Australia
George Nikandros, Australian Safety Critical Systems Association, Australia

Compilation of References 424

About the Contributors 449

Index 460

Detailed Table of Contents

Foreword by Odd Nordland xvii

Foreword by Stephan Jubin xix

Preface xxi

Section 1
Regulations and Certification Standards

Chapter 1
U.S. Regulatory Requirements for Positive Train Control Systems 1
Mark Hartong, Federal Railroad Administration, USA
Duminda Wijesekera, George Mason University, USA

Positive Train Control (PTC) Systems are a type of Communications Based Train Control System (CBTC) designed to enhance railroad safety. As a consequence of a series of high profile train accidents in the United States, a statutory mandate for the installation of these systems in high risk areas by the end of 2015 has been established. This chapter identifies the impetus behind the statute, the statutory requirements associated with PTC, the implementing regulations for the statutory requirements, and the current status of regulatory compliance.

Chapter 2
The Model-Driven openETCS Paradigm for Secure, Safe and
Certifiable Train Control Systems 22
Jan Peleska, University of Bremen, Germany
Johannes Feuser, University of Bremen, Germany
Anne E. Haxthausen, Technical University of Denmark, Denmark

A novel approach to managing development, verification, and validation artifacts for the European Train Control System as open, publicly available items is analyzed and discussed with respect to its implications on system safety, security, and certifiability. After introducing this so-called model-driven openETCS approach, a threat analysis is performed, identifying both safety and security hazards that may be common to all model-based development paradigms for safety-critical railway control systems, or specific to the openETCS approach. In the subsequent sections state-of-the-art methods suitable to counter these threats are reviewed, and novel promising research results are described. These research results comprise domain-specific modeling, model-based code generation in combination with automated object code verification and explicit utilization of virtual machines to ensure containment of security hazards.

Section 2
Hazard Analysis and Model-Based Evaluation

Chapter 3
Semi-Quantitative Risk Assessment of Technical Systems on European Railways 54
Jens Braband, Siemens AG, Germany

The European Railway Agency (ERA) has the challenging task of establishing common safety targets and common safety methods throughout Europe. In this context, the harmonization of risk analysis methods is also discussed. The purpose of this paper is to present a new semi-quantitative approach for the risk analysis of technical systems and the means by which compliance with legal and regulatory requirements can be demonstrated. As a particular reference, a new German pre-standard, which lays out requirements for semi-quantitative approaches, is taken into account.

Chapter 4
The ForMoSA Approach to Qualitative and Quantitative Model-Based Safety Analysis 65
Axel Habermaier, Universität Augsburg, Institut für Informatik, Germany
Matthias Güdemann, Otto-von-Guericke University of Magdeburg, Germany
Frank Ortmeier, Otto-von-Guericke University of Magdeburg, Germany
Wolfgang Reif, Universität Augsburg, Institut für Informatik, Germany
Gerhard Schellhorn, Universität Augsburg, Institut für Informatik, Germany

This chapter presents ForMoSA (FORmal MOdels and Safety Analysis), an integrated approach for the safety assessment of safety-critical embedded systems. The approach brings together the best of engineering practice, formal methods, and mathematics: traditional safety analysis, temporal logics and verification, as well as statistics and optimization. These three orthogonal techniques cover three different aspects of safety: fault tolerance, functional correctness, and quantitative analysis. The ForMoSA approach combines these techniques to assess system safety in a structured and formal way. Furthermore, the tight combination of methods from different analysis domains results in mutual benefits. The combined approach yields results which cannot be produced by any single technique on its own. The methodology was applied to several case studies from different industrial domains. One of them is an autonomous control of level crossings using radio-based communication, which is used in this chapter to describe the individual steps of the ForMoSA methodology.

Section 3
Verification and Validation

Chapter 5
Verification and Validation of Interoperability 116
Lars Ebrecht, DLR (German Aerospace Center), Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, DLR (German Aerospace Center), Institute of Transportation Systems, Germany

The chapter shows an approach to use existing test methods to prove technical as well as operational interoperability. The first kinds of tests are test sequences to validate conformity of a single constituent – here, an on-board on-board unit (OBU) of the European Train Control System (ETCS) in the European Rail Traffic Management System (ERTMS). The second kind of tests is the integration test for assemblies – here, the complete on-board equipment. The third kinds of tests are the tests for the

validation of operational serviceability. An approach for the stepwise integration of the different kinds of tests is shown. As a conclusion the perspective for the use of these test sequences in an independent test lab is given.

Chapter 6

Fault Injection for On-Board ERTMS/ETCS Safety Assessment 128
Almir Villaro Arriola, CEIT and Tecnun (University of Navarra), Spain
Jon Mendizabal Samper, CEIT and Tecnun (University of Navarra), Spain
Juan Meléndez Lagunilla, CEIT and Tecnun (University of Navarra), Spain

On-Board ERTMS/ETCS equipment performs safety related functions where the tolerable hazard rate is kept below 10-9 f/h. Safety standards such as EN50129 or IEC61508 impose requirements on the architecture used to fulfill this safety figure and the associated Safety Integrity Level (SIL). From these standards, the mandatory use of redundancy and physical independence can be derived. Due to the introduction of these requirements, a new functionality is added at the system level (e.g. majority voting processes among redundant lines). Unfortunately, neither the safety nor the interoperability standards provide technical specification that defines how to test the performance of the complete system when internal malfunction has occurred in safety related components. This chapter proposes the use of fault injection techniques to facilitate safety assessment. By means of communication saboteurs, it is possible to excite and test the associated internal functionality in systems performing safety related functions. The chapter also contributes to the definition of the test setup and test procedure of the architecture-associated safety-related internal functionality of the SIL4 odometer and Balise Transmission Module (BTM) subsystems within the on-board European Railway Traffic Management System/ European Train Control System (ERTMS/ETCS).

Chapter 7

Impact of Electromagnetic Environment on Reliability Assessment for Railway Signalling Systems 151
Iñigo Adin, CEIT and Tecnun (University of Navarra), Spain
Jaizki Mendizabal, CEIT and Tecnun (University of Navarra), Spain
Jon del Portillo, CEIT and Tecnun (University of Navarra), Spain

The electromagnetic interferences (EMI) are threats that affect the reliability of the railway signalling systems. Consequently, the identification of the reliability requirements dependent on environment conditions is a major issue for signalling systems designers, and therefore for evaluators, and testing and certification bodies. Signalling systems work in the complex and heterogeneous railway environment, where low power electronics have to work together with high voltages and currents from trains and railway infrastructure. This chapter presents the relationship between the railway electromagnetic interoperability and the reliability assessment by analyzing the signalling systems and the associated inter-dependencies with other components of the rolling stock. It is composed of two main sections; the first gathers an exhaustive state of the art approach to the issue of electromagnetic interoperability and railway industry. This subsection steers towards the combination of electromagnetic interferences and the signalling systems present in the rolling stock noise environment. That is the basis of the second section that finally sets how to establish the reliability requirement for a communication path in this environment. This requirement is established because of the electromagnetic noise environment, as well as the radiated and conducted fields, which are a combination of all the surrounding threats a focused railway system has to face. It also depends on the modulation of the communication signal under study.

Section 4
Automation in Development and Testing

Chapter 8
Μίνθα: A Framework for Auto-Programming and Testing of Railway Controllers for Varying Clients 175
Jörn Guy Süß, University of Queensland, Australia
Neil Robinson, RGB Assurance, Australia
David Carrington, University of Queensland, Australia
Paul Strooper, University of Queensland, Australia

Implementation of railway controller application logic is a highly safety-critical and time-consuming task carried out individually for each client and station by specialised signalling engineers, with corresponding high costs. Μίνθα is a software development framework designed to create code generators for application logic for the client railway companies of Ansaldo STS that use the Microlok II controller to lower the cost and increase repeatability. This chapter describes the evolution of Μίνθα from prototype to framework, and introduces the software engineering approaches of object-oriented meta-modelling and framework development along the way. It also presents known limitations and further application areas of the framework.

Chapter 9
Software-Based Self-Test for Reliable Applications in Railway Systems 198
Alfredo Benso, Politecnico di Torino, Italy
Stefano Di Carlo, Politecnico di Torino, Italy
Alessandro Savino, Politecnico di Torino, Italy

The very strict safety standards, which must be guaranteed in a railway system, make the testing of all electronic components a unique and challenging case study. Software-based self-test represents a very attractive test solution to cope with the problem of on-line and off-line testing of microprocessor-based systems. It makes it possible to deeply test hardware components without introducing extra hardware and stressing the system in its operational condition. This chapter overviews the basic principles of software-based self-test techniques, focusing on a set of best practices to be applied in writing, verifying and computing the final test coverage of high-quality test programs for railway systems.

Chapter 10
Real-Time Hardware-in-the-Loop in Railway: Simulations for Testing Control Software of Electromechanical Train Components 221
Silvio Baccari, University of Sannio, Italy
Giulio Cammeo, AnsaldoBreda, Italy
Christian Dufour, Opal-RT Technologies, Canada
Luigi Iannelli, University of Sannio, Italy
Vincenzo Mungiguerra, AnsaldoBreda, Italy
Mario Porzio, AnsaldoBreda, Italy
Gabriella Reale, University of Sannio, Italy
Francesco Vasca, University of Sannio, Italy

The increasing complexity of modern ground vehicles is making crucial the role of control for improving energetic efficiency, comfort and performance. At the same time, the control software must be frequently updated in order to let the vehicle respond safely and efficiently within more sophisticated environments and to optimize the operations when new vehicle components are integrated. In this framework real-time hardware-in-the-loop simulations represent a fundamental tool for supporting the verification and validation processes of the control software and hardware. In this chapter a railway case study will be presented. The mathematical models of the most relevant electromechanical components of the vehicle powertrain are presented: the pantograph connected to an ideal overhead line with continuous voltage; the electrical components of a pre-charge circuit, the line filter and the braking chopper; the three-phase voltage source inverter and the induction motor; and, finally, the mechanical transmission system, including its interactions with the rail. Then the issues related to the real-time simulation of the locomotive components models are discussed, concentrating on challenges related to the stiff nature of the dynamic equations and on their numerical integration by combining field programmable gate array (FPGA) and central processing unit (CPU) boards. The usefulness of the real-time hardware-in-the-loop simulations for the analysis of railway control software will be demonstrated by considering the powertrains of two real metropolitan trains under complex scenarios, i.e., stator winding disconnection of the induction motor, pantograph missing contact, wheel-rail slipping phenomenon.

Chapter 11
Hardware-In-the-Loop Testing of On-Board Subsystems:
Some Case Studies and Applications 249
Luca Pugi, University of Florence, Italy
Benedetto Allotta, University of Florence, Italy

Hardware In the Loop testing is a very powerful tool for the development, tuning, and synthesized homologation of safety-relevant on-board subsystems and components. In this chapter some case-studies, based on typical topics of industrial research for railways, are introduced in order to emphasize some aspects of the mechatronic design with a particular attention to the integration of actuation systems into rig design.

Section 5
Formal Methods

Chapter 12
The Role of Formal Methods in Software Development for Railway Applications 282
Alessandro Fantechi, Università degli Studi di Firenze, Italy

Formal methods for thirty years have promised to be the solution for the safety certification headaches of railway software designers. This chapter looks at the current industrial application of formal methods in the railway domain. After a recall of the dawning of formal methods in this domain, recent trends are presented that focus in particular on formal verification by means of model checking engines, with its potential and limitations. The paper ends with a perspective into the next future, in which formal methods will be expected to pervade in more respects the production of railway software and systems.

Chapter 13
Symbolic Model Checking for Interlocking Systems 298
Kirsten Winter, The University of Queensland, Australia

Model checking is a fully automated technique for the analysis of a model of a system. Due to its degree of automation it is in principle suitable for application in industry but at the same time its scalability is limited. Symbolic model checking is one approach that improves scalability through the use of Binary Decision Diagrams (BDDs) as an internal data structure. This approach allows the user to increase the efficiency by customising the ordering of state variables occurring in the model to be checked. In the domain of railway interlockings represented as control tables, it is found that this task can be supported using an algorithm that has access to the track layout information. In our work we propose optimisation strategies that render symbolic model checking feasible for large scale interlocking systems. Our results yield a verification tool suitable for use in industry.

Section 6
Human Factors

Chapter 14
Designing Usable Interactive Systems within the Railway Domain:
A Human Factors Approach 317
Nina Jellentrup, German Aerospace Center, Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, German Aerospace Center, Institute of Transportation Systems, Germany

Train drivers as well as signallers interact with several computer based information and communication systems to ensure safe and effective train operations. So far the technical progress mostly determines the design of such interactive systems and requirements out of a human factors perspective are not integrated. Beside the development of technical functions it is essential to take the usability as a quality attribute of every interactive system into account. If the usability is not considered during system development, it could occur that there are several functions available within a system but the user does not know how to use them in an efficient way. This chapter describes a psychological approach to design or redesign usable interactive systems within the railway domain. Some examples will be discussed to demonstrate the approach and the results.

Chapter 15
Integration of Human Factors to Safety Assessments by Human Barrier Interaction 327
Markus Talg, German Aerospace Center, Institute of Transportation Systems, Germany
Malte Hammerl, German Aerospace Center, Institute of Transportation Systems, Germany
Michael Meyer zu Hörste, German Aerospace Center, Institute of Transportation Systems, Germany

Human factors have a strong impact on railways safety. However, the assessments of these factors still follow traditional and inadequate approaches. While failure probabilities of technical systems can be measured in sufficient precision, human error probabilities are still estimated in a very rough and vague way. Upon this motivation, the contribution presents a method analyzing human influence in railway applications. The approach of human-barrier-interaction relies on a new model of human behavior, a classic model of human-machine-interaction and a model of safety measures by barriers. Applying the method, human reliability can be assessed in comparative way. An advantage over existing approaches is the substantial combination of cognitive psychology and engineering expertise without unpractical complexity.

Section 7
Security, Monitoring and Surveillance

Chapter 16
Advanced Techniques for Monitoring the Condition of Mission-Critical Railway Equipment 341
Clive Roberts, University of Birmingham, UK
Joe Silmon, University of Birmingham, UK

This chapter provides an overview of advanced techniques for monitoring the condition of mission-critical railway assets. The safe operation of railways depends on a large number of geographically distributed components, each of which has a low cost when compared to the highly complex arrangements of assets found in other industries, such as rolling mills and chemical plants. Failure of any one of these components usually results in a degradation of service in order to maintain safety, and is thus very costly to modern railway operators, who are required to compensate their customers when delays occur. In this chapter, techniques for industrial condition monitoring are reviewed, highlighting the main approaches and their applicability, advantages, and disadvantages. The chapter first makes some basic definitions of faults, failures, and machine conditions. The analysis of faults through methods such as Fault Tree Analysis and Failure Modes Effects Analysis are examined. The field of fault diagnosis is then reviewed, partitioning into the three main areas: numeric/analytical models, qualitative models, and data/history-based methods. Some of the key approaches within each of these areas will be explained at a high level, compared, and contrasted.

Chapter 17
Security of Railway Infrastructures 355
A. Di Febbraro, University of Genoa, Italy
F. Papa, University of Genoa, Italy
N. Sacco, University of Genoa, Italy

In recent years, some sadly famous terrorist attacks that occurred in different countries have put into evidence that railway transportation systems are not suitably protected, and not capable of tolerating and promptly reacting to them. Moreover, it is clear that such mass transportation systems are particularly attractive for terrorists, due to the potentially far-reaching, often "spectacular" results of attacks. Examples of such kinds of events are the New York (2001), Madrid (2004), and London (2005) terrorist attacks. In addition, by focusing on ground transportation networks, and especially on railway systems, it is also easy to observe that they are particularly difficult to be secured since they are characterized by high accessibility and wide extension, as also noted by Fink (2003). In this sense, the needs of security and of mobility often conflict with each other. In effect, while an open and accessible system provides an efficient transportation of people and goods, this openness also allows malicious entities to exploit the transportation system as a target, weapon, or means to reach another target (Murray-Tuite, 2007). Then, on the contrary, it is clearly evident that security actions taken to limit malicious adversaries from reaching or capturing their targets may degrade the transportation system performances, so they have to be designed with particular attention. This is the reason why worldwide institutions are more and more sensitive to the growing need for security of the so-called Critical Infrastructures (CI), such as railway transportation systems, and are adopting a number of regulatory measures (US Congress, 2007; EU Commission, 2005, 2008, and 2010). For what concerns scientific research, the efforts are intended to define methodologies, build risk mitigation devices, and find out best practices that are technologically advanced, soon achievable, reliable, so as to increase the infrastructure protection without affecting the relevant transportation system performances. In this framework, Quantitative Risk Analysis (QRA) represents the main methodological approach for assessing security, which is indeed often character-

ized by a large set of variables dependent on human sensitivity, and requires calibration and adaptive tuning, thus resulting into unfriendly tools for the non-skilled users. Then, in this chapter, to tackle with the problem of clarifying the aims, the characteristics, and the limitations, a general architecture for a possible QRA tool for railway security assessment is presented, with particular attention to the relevant specifications (Di Febbraro et al., 2010).

Section 8
Experiences and Case-Studies

Chapter 18

ETCS Developing and Operation: Italian Experience 381
Raffaele Malangone, RFI, Italy
Fabio Senesi, ANSF, Italy

ETCS/ERTMS actually is the present for Italian Railways but it will also be the next future for the signalling system in many countries and the best technological choice for ATC (Automatic Train Control) systems. Italian Railways, first in the world, have carried out ETCS Level2 merging the technologies and the regulations respecting the highest safety level. RFI, following the CENELEC 50-126 Lyfe-Cycle, has developed a process for planning, managing, monitoring and controlling of ETCS achievements. In particular the Disposal 29 and 32 in the year 2002 have been issued for the assessment and homologation process of Generic and Specific Applications and other following procedures have permitted the final configuration of the project until its putting in service. The goal of this course is the Preliminary Acceptance of a Generic Application and later, after a successful testing period on field, its Homologation. RFI has followed the developing process starting from the idea to define the specifications, evaluating the hazards and their probabilities, finding mitigations to improve the safety, validating the products of the suppliers, testing the subsystems and the entire system, until the final activation of the whole systems (compliant with the Technical Specifications for interoperability, UNISIG v.2.2.2).Much attention has been paid to the testing of functional scenarios (using also formal languages) and the real tests on the track have been reduced with the support of an ERTMS Laboratory in Rome (unique in the world for its characteristics) where on-board and track-side subsystem permit to reproduce easily and quickly most of the real situations. This testing process in ETCS laboratories has been useful not only before the putting the ETCS in service but also for the reconfiguration of the actual ETCS lines as it would be hard to do so many test scenario during a commercial service. These activities have been replicated several times, for example, to reach the actual ETCS version compliant to the UNISIG 2.3.0d.The success of the formal language analysis of Test-Specifications has also encouraged the RFI ETCS group to develop a state-charts model of the functional specification. This work is actually in progress but a first result, on the logical behavior of the system at the transition with a historical signalling system, has been done and validated.

Chapter 19

Adoption of Low-Cost Rail Level Crossing Warning Devices: An Australian Case Study 399
Christian Wullems, Cooperative Research Centre for Rail Innovation, & Centre for Accident Research and Road Safety – Queensland (CARRS-Q), Australia
George Nikandros, Australian Safety Critical Systems Association, Australia

The objective of this chapter is to provide rail practitioners with a practical approach for determining safety requirements of low-cost level crossing warning devices (LCLCWDs) on an Australian railway by way of a case study. LCLCWDs, in theory, allow railway operators to improve the safety of passively

controlled crossing by upgrading a larger number of level crossings with the same budget that would otherwise be used to upgrade these using the conventional active level crossing control technologies, e.g. track circuit initiated flashing light systems. The chapter discusses the experience and obstacles of adopting LCLCWDs in Australia, and demonstrates how the risk-based approach may be used to make the case for LCLCWDs.

Compilation of References 424

About the Contributors 449

Index 460

Foreword

by Odd Nordland

Rail transportation, in the widest sense of the expression, is becoming more and more significant both economically and ecologically. The growing number of megacities need mass transportation systems both within their ever expanding city limits, but also between the urban centres. In addition, more isolated areas need cost efficient, safe, and reliable transportation systems for people and goods into the urban centres. Politicians all over the world are beginning to realise the importance of a well-developed and stable mass transport infrastructure as an alternative to the ever expanding and accident prone individual transportation on congested roads, as is witnessed by the numerous strategic programs for railway and metro systems.

The introduction of the European Rail Traffic Management System ERTMS some 20 years ago has led to a rejuvenation of railways within Europe. This process has been reinforced by the EU directives on interoperability that started with high-speed railways, continued with conventional rail and have now been united to a single directive for both high-speed and conventional rail. The directive aims at achieving harmonised railway systems that can traffic throughout Europe without encountering national barriers. The ERTMS technology is being adopted outside Europe so that the day will come when "orient express" will mean trains travelling from Paris to Beijing!

The processes for authorising the use of railway systems are also being harmonised in Europe and work is being done to define common safety requirements and common safety methods for use with ERTMS in the European railways. ERTMS is a communication based train control system not unlike the systems that are widespread in urban and light rail around the world, and the methods and techniques that are emerging for European interoperability will be equally applicable to urban and light rail systems all over the world. Just as the ERTMS technology is being adopted also outside Europe, we can expect that common European safety requirements and methods will be adopted by the international community.

This book presents state of the art techniques that are actually being used in real life applications by leading experts in the field. It contains descriptions and explanations of methods and techniques that have been developed in order to achieve or improve the dependability of existing and projected railway applications. As such it is not only important for students and professionals who want to learn and apply the best practices of today; it is also a compendium of techniques that can be approved by assessors and notified bodies.

Technology is developing rapidly, and with it there are emerging new requirements requiring innovative solutions. This applies not only to the technical requirements, but also to the methods and processes that are applied. The techniques for demonstrating not only reliability, availability, maintainability and safety, but also interoperability must be suitable for the new technologies. Techniques that a few years ago were considered to be "exotic" in a railway context are now becoming more and more established, and this book gives a good insight into the latest developments.

The inclusion of security aspects deserves particular mention. For decades, safety and security have been disjoint worlds with an almost hostile relationship to each other, but experts on both sides are moving towards each other. It is now accepted that safety without security is wishful thinking, and that security without safety is a contradiction.

Whether you are a student who wants to learn, a practitioner wants to apply or an assessor who needs to evaluate state of the art techniques, this book will provide you with the information you need.

Odd Nordland
SINTEF ICT Centre for Railway Certification, Norway

Foreword

by Stephan Jubin

Dear Reader,

It is my pleasure to welcome you to the introduction of this book. Selecting this book as your current lecture shows that we have a common interest: safety of railway systems.

Indeed railway systems are facing huge challenges in the last years. We are observing some significant trends: one is towards higher operational speeds; another one is towards highly or even fully automated systems. Both trends include the need for an increased usage of programmable electronic systems to support the Operators in their duty; or in case of fully automated systems even to replace some staff. But also the conventional railway, metro and tram systems make more and more use of electronic systems, especially with the expectation to increase operational effectiveness and customer satisfaction – and thus to optimise revenue success.

And indeed the introduction of electronic systems seems to provide a lot of advantages: computers appear as less error prone than human operators, they provide additional and more flexible functionality, passenger information and comfort is improved, and also the operational performance benefits in terms of (shorter) headways and improved punctuality.

Unfortunately, as nearly always, there is the little word "but" – also in this case. It is related to the fact that computer systems just tend to appear less error prone, but in fact this assumption only holds if some very specific care is applied. This needs to start already in the beginning of the design in order to come to a careful decision which functions to automate and which not. In this context not only the electronic system alone can be considered; the whole function needs to be in the focus including all the sensors to obtain the necessary information and all the actuators to control/perform the function. Of course it is an excellent idea to avoid human errors in the operation of a railway system. But transfer of operational responsibility from humans to technical equipment is not always the best solution since humans also have their advantages. In particular the human sensory organs, which are able to recognize various different situations within very short time and the high intellectual flexibility to react on the scenarios can be reflected by technical solutions only to a limited degree. However, if an appropriate technical representation is possible, then the technical solution in most cases shows the more reliable performance.

But how to perform such safety analyses?

How to determine the safety relevance of the product you are going to design?

Which means can be applied to ensure that this product will perform sufficiently safe?

How to demonstrate the safety capabilities of your product towards other parties such as clients or Safety Authorities?

This is now the point where this book enters the scene. Here several renowned experts from leading railway suppliers, universities and research institutes have shown-up to share their experience. You will

find helpful ideas and guidance on how specific safety matters can be addressed. This covers safety and risk analyses, the consideration of human factors, and several detailed reports about how the safety of an electronic product can be demonstrated. Included are approaches which are already well-proven from past projects as well as new ideas to deal with future challenges. Due to their special safety relevance there is by nature a focus on signalling and train protection systems. However the described concepts might be applicable also for other safety relevant systems or components, too.

I would like to warmly invite you to the lecture of the following sections, to benefit from the experience and ideas of the various experts, and based on the gained impressions I wish you many good ideas for the further development and improvement of your own products and railway systems.

Yours sincerely

Stephan Jubin
TÜV Rheinland InterTraffic GmbH ACR – Assessment & Certification Rail, Germany

Preface

The pervasiveness and efficiency of rail-based transportation infrastructures is generally and justifiably perceived as one of the most evident signs of civilization of a territory. In fact, those systems are able to move a large mass of people through urban or country areas with a high level of safety with respect to road transportation, and a reduced environmental impact due to electrical traction and possible underground operation.

In order to make the safety of rail transportation less dependent on human supervision, that is unfortunately error-prone as witnessed by the many tragic accidents happened in the (even recent) past, computer-based control systems are being increasingly adopted for automatic train protection and operation. In the most advanced of those systems, radio-signalling is employed, allowing on-board systems to receive information about their movement authority and speed profiles using data packets sent over wireless networks, like GSM or Wi-Fi (e.g. in CBTC, Communication Based Train Control). We are talking about control systems which belong to the category known as 'real-time safety-critical', since information integrity and timeliness is essential to prevent catastrophic failures, that is serious consequences on the health of human-beings and/or on the environment. Also, mass-transit systems, railways, and metros are attractive targets for criminals and terrorists, and therefore need to be protected even against intentional threats.

In such a context, this book provides engineering students and professionals with a collection of state-of-the-art methodological and technological notions to support the development and certification of railway control systems as well as the protection of rail transportation infrastructures.

To that aim, this book surveys the following main topics in the railway transportation domain:

1. **Regulations and Certification Standards**: Covering both Positive Train Control (PTC) and European Train Control System (ETCS).
2. **Hazard Analysis and Model-Based Evaluation**: Including qualitative, semi-quantitiative and quantitative approaches to risk and dependability assessment.
3. **Verification and Validation**: Including interoperability testing, fault-injection and electromagnetic testing.
4. **Automation in Development and Testing**: Addressing frameworks for auto-programming and self-testing.
5. **Formal Methods**: For the verification of railway control logics and the symbolic model-checking of interlocking systems.
6. **Human Factors**: Addressing the usability of machine interfaces and the study of human-barriers.
7. **Security, Monitoring and Surveillance**: Whose concepts and technologies are applied to mission-critical railway equipment and transportation infrastructures.

8. **Experiences and Case-Studies**: Addressing some relevant development and maintenance applications of level-crossing and train control systems.

I believe the topics listed above represent a comprehensive picture of engineering best-practices in the railway domain. As such, they should be part of the technical background of engineers involved in the reliability, safety, and security assurance of railway control systems and infrastructures.

Finally, I would like to thank all the outstanding researchers and professionals who have decided to invest part of their valuable time to contribute to the preparation and to the review of this manuscript.

Dr. Francesco Flammini
Book Editor

Section 1

Regulations and Certification Standards

Chapter 1
U.S. Regulatory Requirements for Positive Train Control Systems

Mark Hartong
Federal Railroad Administration, USA

Duminda Wijesekera
George Mason University, USA

ABSTRACT

Positive Train Control (PTC) Systems are a type of Communications Based Train Control System (CBTC) designed to enhance railroad safety. As a consequence of a series of high profile train accidents in the United States, a statutory mandate for the installation of these systems in high risk areas by the end of 2015 has been established. This chapter identifies the impetus behind the statute, the statutory requirements associated with PTC, the implementing regulations for the statutory requirements, and the current status of regulatory compliance.

INTRODUCTION

Railroads can be found on every continent on the globe except Antarctica carrying passengers and freight. Depending on the country they operate in, railroads are overseen by a variety of governmental, quasi-governmental, and non-governmental agencies. In the United States (U.S.), railroads are regulated by two different agencies of the U.S. Federal government. The first is the Surface Transportation Board (STB), which regulates railroad rates, resolves service disputes, railroad mergers, sales, construction and abandonments. The second is the Federal Railroad Administration (FRA) whose primary function is to promulgate and enforce rail safety regulations. Among the various types of railroad safety systems that are governed by FRA regulation is Positive Train Control (PTC), a form of Communication Based Train Control (CBTC). This chapter will discuss

DOI: 10.4018/978-1-4666-1643-1.ch001

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

PTC in the U.S., its history, and U.S. federal government regulations regarding it use.

The other most significant PTC regulatory efforts undertaken to date worldwide, specifically the European Commission and the European Railway Agency (ERA) efforts with the European Train Control System (ETCS), have been driven to a large extent by the objective of facilitating transnational interoperability by replacing a large number of different existing national train control systems installed across the overwhelming majority of the European rail network with a single harmonized system. The result of the ERA efforts has been the development of a series of increasingly detailed Technical Interoperability Standards (TIS) that define not only the train control system and infrastructure, but the communications infrastructure as well. (EC, 2010)

This differs significantly from the US situation. On roughly 60% of the U.S. rail network, there are no train control systems of any type deployed, and verbal authorities passed to the train crew from the dispatcher control train movements. Additionally, unlike Europe, the US railroads are by and large privately owned and are vertically integrated, owning and operating not only the rolling stock and but the underlying infrastructure. As a consequence the US regulatory efforts have not focused on detailed specifications of technical interoperability requirements, but rather on the general performance objectives that any system, regardless of the underlying technology, must meet. Individual U.S. railroads and their suppliers determine the detailed technical requirements for the systems they elect to procure and install as opposed to the federal government.

BACKGROUND

Rail operations in the U.S. are predominately freight, with comparatively limited passenger/commuter services. The U.S. rail system is the largest integrated freight and passenger system in the world, operating over 169,000 miles of track (AAR, 2010; IUC 2010). The 563 freight railroads employ almost 170,000 people and operate over 1.2 million freight cars. The 563 freight railroads are dominated by seven "Class 1"[1] railroads; the Burlington Northern and Santa Fe (BNSF), CSX, Kansas City Southern (KCS), Norfolk Southern (NS) Canadian Pacific (CP - Soo Line), Canadian National (CN- Grand Trunk) and Union Pacific (UP). The CP (Soo Line) and CN (Grand Trunk) represent the U.S. operations of the CP and CN Railroads respectively.

The Class 1 freight railroads play a significant role in the U.S. economy. They moved over 1,820 billion ton miles of freight, or 42% of all US freight traffic by revenue mile (DOT, 2007). In contrast, the 27 members of the European Union only moved 447 billion tonne-kilometers, 18.1% of all freight traffic by tone-kilometer (EU, 2010). In terms of operating revenue, the 2009 aggregate revenues of the BNSF, UP, CSX, and NS alone were over $44 billion dollars. If these numbers are viewed as a country, then their combined revenues would rank them as the 72st largest country in the world by Gross Domestic Product (GDP). (IMF, 2010)

Passenger/commuter service in the U.S. is modest, especially when compared to passenger/commuter services elsewhere in the world. The National Railroad Passenger Corporation (Amtrak) provides long distance intercity service. While providing service to 46 states and the District of Columbia, it only has a 22,000 mile network the majority of which is over freight railroad tracks. If included among U.S. airlines in 2008, Amtrak would only rank 8th in the number of passengers served, with only 6.6 Billion revenue passenger miles (FRA, 2010; DOT, 2010). Local commuter service is provided by 23 local agencies. Operating over 8,000 miles of track, they had only slightly over 11 billion revenue passenger miles[2] (APTA, 2010). By comparison, the 27 members of the EU had almost 399 billion revenue passenger kilometers (EU, 2010), or more than 12 times the combined U.S. intercity and commuter passenger miles.

Despite the immense scope of U.S. rail operations, passenger and freight rail transportation on a per-mile basis is exponentially safer than cars or truck transportation. For example, in 2008 over 6 million freight accidents occurred on highways compared to rail, which accounted for just over 5,000 accidents. Between 2003 and 2007, rail transport averaged .39 fatalities per billion ton miles, compared to 2.55 fatalities per billion ton miles for truck transport. For the same time period fatalities in passenger cars occurred at a rate of 7.97 per billion passenger miles as compared to intercity passenger rail with a rate of 1.08 per billion passenger miles. When considering the movement of hazardous material, U.S. freight railroads transported more than 99.99 percent of rail hazmat shipments to their destination without a release caused by a train accident. Rails share of hazardous materials transport was 20 percent of all U.S. chemicals moves (DOT, 2010).

Although U.S. railroads overall have had an excellent safety record, railroad accidents when they occur have tended to be "newsworthy". Two such accidents, which were to have a significant impact on U.S. public policy with respect to PTC, were a collision between two NS freight trains in Graniteville, South Carolina (NTSB, 2005) and a collision between a UP freight train and a Metrolink commuter train in Chatsworth, California (NTSB, 2010).

Graniteville, SC Collision

Graniteville is a small town of 7000 people located in the Central Savannah River Area of South Carolina. On January 6, 2005 around 0240 hours, a Norfolk Southern Railway traveling on the mainline in non signaled territory struck an unoccupied parked train in an industrial siding at 42 miles per hour and derailed after encountering a misaligned switch. The resulting collision resulted in the breech of a tank car containing 180,000 lbs of chlorine. The release resulted in nine fatalities, 554 injures and necessitated in the evacuation of about 5,400 people in a 1 mile radius of the collision site. The post accident investigation by the National Transportation Safety Board (NTSB) and the FRA determined that the cause of the accident was the failure of the crew of the parked train to properly realign and lock the switch for the mainline when they went off duty the previous evening. Contributing to the accident was the absence of any feature or mechanism that would have reminded crewmembers of the switch position.

The failure of train crews to properly lock and line switches is not uncommon. During the period January2001 to December 2004, over 47% of accidents required to be reported to FRA were "human performance" failures. Of the 47% of human performance related accidents, over 18% were switches not properly lined or locked (FRA, 2010). After the Graniteville accident, FRA issued a Safety Advisory to advise all railroads to review their crew training and operating rules and take necessary action to ensure that train crews who operate manual (hand-operated) main track switches in non-signaled territory restore the switches to their normal position after use. Subsequent to the Safety Advisory, FRA issued an Emergency Order mandating additional training and implementation of additional administrative controls regarding switch operations.(FRA, 2005) However given the extremely adverse cost benefit ratio (over 20:1 against) associated with implementation of PTC, it did not meet the necessary criteria for FRA to mandate the installation of PTC.

Chatsworth, CA Collision

Roughly three years later the final impetus that forced a change in U.S. public policy occurred. On September 12, 2008 a Southern California Regional Rail Authority (SCRRA) Metrolink commuter train collided head-on with a UP freight train near Chatsworth California. Metrolink provides commuter rail serve in and around the city of Los Angles. Chatsworth, the site of the accident, is a suburb of Los Angles, is located northwest

of the city in the San Fernando Valley. The accident occurred around 1620 hours head-on at a combined speed of over 80 miles per hour. It was daylight, with clear skies and 4 miles of visibility. The resulting collision caused the locomotive the Metrolink train to telescope back into the lead passenger coach by about 52 feet with 25 fatalities and 135 injures. It was the worst train accident in US history since 1993 when the Amtrak "Sunset Limited" derailed into the Alabama's Big Bayou Canot killing 47 people (NTSB, 1993)

The NTSB and FRA post accident investigation revealed the engineer of the Metrolink train had been "texting" on his cell phone immediately before the accident and had made no change in throttle position or brake application. On the day of the collision, during the time the Metrolink engineer was responsible for the operation of the train, the engineer sent 21 text messages, received 21 text messages, and made four outgoing telephone calls. The Metrolink engineer ran through a flashing advanced yellow signal, a solid yellow signal intermediate signal indicating the next signal ahead was a red stop, the red stop signal and through a switch that had been aligned for the UP freight train movement. The accident at Chatsworth was the second accident involving a collision between a freight train and a Metrolink passenger train that the NTSB had investigated. In a 2002 collision in Placentia, California, a BNSF freight train failed to comply with an approach signal indication and was therefore unable to stop short of the next signal, which was displaying a stop indication (NTSB, 2003). The BNSF train continued past the stop signal and collided head-on with a Metrolink passenger train. That earlier accident resulted in 2 fatalities and more than 100 injuries.

POSITIVE TRAIN CONTROL IN THE US

A common finding by the NTSB in both the Graniteville and Chatsworth Accidents, as well as a number of earlier less public rail accidents, was that the accidents could have been prevented if the trains had been equipped with PTC.` PTC describes technologies designed to automatically stop or slow a train before certain accidents caused by human error occur — specifically, train-to-train collisions, derailments caused by excessive speed, unauthorized incursions by trains onto maintenance work zones, and movement of a train through a switch improperly aligned for the train movement.

A Short History

The origins of PTC in the US are generally traced to 1981, when Richard Bressler, the then new chairman of the Burlington Northern (BN) railroad became aware of a new generation of avionics being installed in commercial aircraft. The avionics included an "energy management system" that advised the flight crew on optimal throttle settings to maximize fuel economy. Fuel prices in the US at that time were high in the wake of the Iranian revolution and Bressler asked his R&D staff if the technology had any application to locomotives. BN research staff believed it could, and began development of the Advanced Railroad Electronics System (ARES) with Rockwell International as the system integrator. The system was developed, and successfully tested as a working prototype in Minnesota's Iron Range, for five years in the 1980s (Strauss, 1998)

The original justification for ARES was accident prevention and fuel economy. However, it became apparent that the cost of the system could not be justified by savings in these areas. In 1988, a team of contractors was assembled by BN to build a "business case" for ARES. Significant potential benefits were identified; however, management was unconvinced and ARES was never used in commercial operations.

The Railway Association of Canada (RAC) and the Association of American Railroads (AAR) initiated the Advanced Train Control Systems

(ATCS) project in 1984, in part as a response to BN's ARES work. The railroads developed an operating requirements document and contracted a team of engineering firms (ARINC Research Corporation, Transportation and Distribution Associates, and Lapp-Hancock, Ltd.) to act as the system engineering team. By 1990, a detailed Control Flows Document was developed, and Canadian National Railways began testing some aspects of ATCS (AAR, 2005). The aim of the ATCS project was to develop a uniform, interoperable control system. However, in 1994 the AAR Board of Directors decided to cease work on ATCS, agreeing that further progress would be up to individual railroads and not an industry-wide effort.

In 1990, the NTSB published its first list of "most wanted" transportation safety improvements. "Positive Train Separation" headed the list. This was later changed to "Positive Train Control Systems" and continued to appear on NTSB's list through 2008 (NTSB, 2008). NTSB recognized that, while safety benefits alone could not justify installation of PTC by the railroads, operational benefits added to safety benefits might make the case.

Partly in response to NTSB, FRA reached agreement with the Illinois Department of Transportation, AAR, and UP in 1998 to install PTC between Chicago, Illinois and St. Louis, Missouri. Lockheed Martin was selected as System Integrator for this project, which had two objectives:

- 110 MPH operation of passenger trains in the corridor, mixed with freight traffic.
- Demonstration of "moving block" [3]technology to increase line capacity by decreasing minimum train headways.

The project was discontinued in September 2006 following repeated failures to solve numerous communications problems (FRA, 2009).

Several other railroads tested various features of PTC during the 1990s and into the 21st Century. BNSF implemented the Electronic Train Management System (ETMS), and CSX tested their Communications-Based Train Management (CBTM) system. Amtrak had successfully implemented Advanced Civil Speed Enforcement System (ACSES) and Incremental Train Control System (ITCS), and obtained FRA's approval of these systems as meeting PTC requirements. BNSF asked for, and received from FRA, a waiver to install its ETMS on 35 subdivisions as an "overlay" system (FRA, 2006). However when the Chatsworth accident occurred in 2008, the railroads still had no plans for a network-wide deployment of PTC despite 20 years of testing, prototyping, and pressure from NTSB and FRA.

The Technology

PTC systems are complex software based systems made-up up of four subsystems (wayside, mobile, communications, and dispatch/control) that have four basic functions, specifically (GPO, 2008):

- Preventing train-to-train collisions.
- Enforcing speed restrictions, including civil engineering restrictions and temporary *slow orders*.
- Protecting roadway workers and their equipment operating under specific authorities.
- Preventing the movement of trains through misaligned switches.

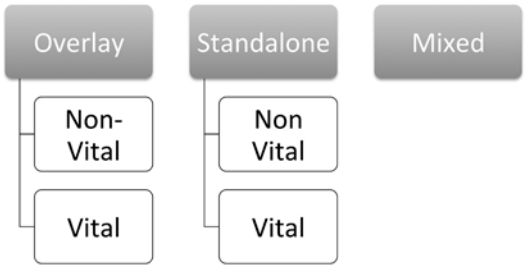
The wayside subsystem consists of elements such as highway grade crossing signals, switches and interlocks or maintenance of wayside workers. The mobile subsystem consists of locomotives or other on rail equipment, with their onboard braking and speed control computers and location determination systems. The dispatch/control unit is the central office that provides the movement authorities to mobile subsystem components or control signals to wayside components. The communication subsystem links the wayside, mobile, and office/dispatch systems together.

System Types

The U.S. federal government classifies PTC systems by the extent that they are used to augment the existing method of railroad operations. Standalone PTC systems completely change, or replace, the existing method of operations. Overlay PTC systems act as a backup to an existing method of operations, which remains unchanged. Overlay systems can be further subdivided based on their design mode of failure. Non-vital PTC overlays are systems that cannot be guaranteed to fail to a safe state. Vital overlays, on the other hand, are designed to be failsafe. Finally there are mixed systems that combine attributes of standalone and overlay systems (GPO, 2010). Although not part of the government classification scheme, PTC systems classifications can be further refined in terms of which of the subsystems caries out the majority of the computational effort to provide the four main PTC functionalities (see Figure 1).

In dispatch/control-based based systems, the dispatch/control subsystem is responsible for the majority of the logical effort required to implement the various PTC functions. The wayside subsystem and mobile subsystem communicate required data to the dispatch/control subsystem through the communications subsystem. The dispatch/control system in the central office aggregates data, analyzes it, and interprets it into the actions necessary to perform the PTC functionality. In wayside-based systems, the wayside devices are responsible for the majority of the logical effort required to implement the PTC functions. Wayside devices aggregate data, analyze it, and interpret the data. Mobile-centric systems behave similarly to dispatch/office based and wayside based systems, but the data is collected analyzed, and interpreted onboard the locomotive or other on track equipment.

Figure 1. PTC System Types



Today in the U.S. there are 8 different domestic PTC systems that implement some variant of these basic PTC system types either in development, under test, or deployed and operational in revenue service on freight and low to medium speed passenger/commuter service. These are as follows.

- Advanced Civil Speed Enforcement System (ACSES),
- Incremental Train Control System (ITCS),
- Electronic Train Management System (ETMS)
- Vital Train Management System (VETMS),
- Collision Avoidance System (CAS),
- Train Sentinel (TS) System,
- North American Joint Positive Train Control (NAJPTC) System, and
- Computer based Train Management (CBTM) System

The PTC system types that will be used for high speed US passenger operations have not yet been determined.

ACSES & ITCS

ACSES is installed and fully operation on U.S. North East Corridor (NEC) between Boston, MA and Washington, DC. Developed to support Amtrak electrified ACELA service at speeds up to 150 miles per hour, ACSES is a track embedded transponder-based system that supplements the exiting NEC cab signal/automatic train control system. Amtrak also operates the ITCS system to support diesel powered passenger operations between Niles and Kalamazoo, Michigan. Unlike ACSES, ITCS is a radio-based system. ITCS is also unique from other

US PTC system implementations in that it includes advanced high-speed highway-rail grade crossing warning system using radio communication rather than track circuits. Depending on the reports received from the Highway Grade Crossing Warning (HGCW) system, the ITCS onboard computer imposes and enforces appropriate speed restrictions. ITCS operations are currently at 95 miles per hour soon to be raised to 110 miles per hour.

ETMS & VETMS

BNSF Railways has undertaken an extensive PTC development and deployment effort to support their freight operations with the non-vital ETMS overlay system. FRA approved ETMS for low-density train operations and BNSF has started deployment on 35 of their subdivisions. BNSF also has an enhanced version of ETMS to support, high-density train and mountain grade train operations awaiting FRA approval, and is actively testing ETMS for passenger/commuter operations in a mixed freight and passenger/commuter-operating environment. The UP, NS and the CSX railroads are developing the V-ETMS PTC system. Although based on the same technology as ETMS, VETMS is intended to be a vital, as opposed to non-vital, overlay PTC system. Because the BNSF, CSX, UP, and NS system proponents, along with KCS, CP (Soo Line) and CN (Grand Trunk) who have also adopted this technology, represent over 90% of the freight railroad operations in the U.S., the ETMS & VETMS systems represent the de-facto standard for PTC in the U.S.

The VETMS and ETMS is are designed and built by WABTEC Railway Electronics and provides for warning and enforcement of speed restrictions (permanent and temporary), work zone boundaries, and route integrity of monitored switches, absolute signals, and track (rail) integrity. During system operation, train crews are notified of potential violations when they are within a sufficient warning distance that allows them to take corrective action. If the crew fails to take corrective action, ETMS applies a full service brake application to stop the train. The method of operations does not change, however, and crews are responsible for complying with railroad operating rules at all times. Both systems share a common software code base, differing only in their specific hardware configurations.

The major components of the VETMS/ETMS mobile subsystem consist of the engineer's color display, a brake interface to connect the control signal output from the train management computer to the brake system, a radio for communications with the wayside and office subsystems, a differential GPS system for location determination and a train management computer with its associated track database of the layout of the railroad. The train crew obtains information using a series of complex graphics on the engineer's display of the track configuration and geometry, switch position, signal indication, authority limits, train direction and makeup, current speeds, max speed, distance to enforcement, time to enforcement, geographical location and text messages. These are augmented by the use of selective color highlighting and audible alarms. The text messages either describe enforcement action in progress, or advise of a condition or required action. In addition, all applicable active warrants and bulletins can be recalled from the onboard database.

The primary means of determining position is via differential GPS, with dead reckoning to augment location determination between GPS fixes. The train management computer continuously compares its received GPS position with the stored position of speed restriction zones, work zones, and monitored switches and signal from the track database in non-volatile memory. As the train management computer determines that the locomotive position is approaching the position of speed restriction and work zones, the train management computer system automatically calculates and activates the brake interface as required. The braking enforcement curves are updated dynamically based on reported changes.

The wayside subsystem consists of a set of wayside interface units that act as a communications front end for switch position, signal indications, and broken rail indications. The mobile subsystem monitors the indications transmitted by the wayside interface units in the train's forward direction of movement. Wayside interface unit s in the wayside subsystem provides the latest state of monitored devices, and the mobile subsystem will accept changes in the indication (with the corresponding changes in required enforcement activity) up to a set distance before reaching the monitored device, after which point a change is ignored.

The communications subsystem consists of a wireless 802.11b broadband network to transfer track database information and event logs at selected access points along the track, and an extended line of sight communications (ELOS) Time division multiplexed network in either the 43 MHz or 220MHz frequency ranges for other data exchange. There is direct continuous exchange of data over the communications subsystem between the wayside and the mobile system, as well as between the mobile and office/control subsystem and the mobile and wayside subsystems.

The dispatch/control subsystem consists of any one of number of different Computer Aided Dispatch Systems (CADS) and a VETMS/ETMS Back Office Server (BOS) for providing train authorities, track data, consist data, and bulletins. Static information, such as track data is stored in the BOS portion of the dispatch/control subsystem, while dynamic information, such as authorities, are stored in the CADS component. Various components of the V-ETMS system are under test on 15 different UP subdivisions in Washington State in the U.S. Pacific Northwest and the Powder River Basin of Wyoming as well as on the NS Charlestown to Columbia SC Subdivisions.

CAS

CAS is a project of the Alaska Railroad (ARRC). The ARRC is in a unique position compared to other railroads in the U.S.. While it supports both freight and passenger operations, as railroads in the lower 48 states, it is not directly interconnected and does not interchange locomotives with any of the other railroads. Freight cars are transferred by barge from Seattle Washington area to Whittier Alaska where they are off loaded and mated up with ARRC locomotives. CAS is being installed on all 531 miles of the ARRC system. CAS is designed as a vital PTC system, and implements the same functions as ETMS and V-ETMS in much the same manner as the V-ETMS/ETMS systems. It is built, however, using completely different and non-compatible hardware and software. CAS enforces movement authority, speed restrictions, and on-track equipment protection in a combination of Direct Traffic Control (DTC) and signaled territory. The wayside subsystem and dispatch/control subsystem have been installed and tested. Mobile subsystem components testing is in progress on the to Portage and Whittier Subdivisions outside of Anchorage, Alaska.

Other Systems

The remaining three PTC system technologies do not have any widespread use in the U.S. TS is currently in use on various railroads in South and Central America. The US version currently being marketed is based on the TS installation currently operating in mixed passenger and high-speed freight service on the Panama Canal Railroad between Balboa and Panama City in the Republic of Panama. The NAJPTC, previously mentioned, was a joint effort of the FRA, the AAR, and the Illinois Department of Transportation to develop an industry open standard high-speed passenger and freight service. Removed from service due to technical issues associated with communications bandwidth, the system was relocated to the US Department of Transportation (DOT) Technology Transportation Center (TTC) Test facility in Pueblo, CO, for study and resolution of the communications issues associated with the standard in a controlled environment where it currently

resides. The CBTM system is under design for the Port Authority of New York and New Jersey (PATH) to provide PTC functionality for underground Trans-Hudson River Commuter Rail Line between New Jersey and New York City.

PTC AS PUBLIC POLICY

Using technology to keep trains apart has long been a strategic public policy goal. The National Transportation Safety Board (NTSB) made its first recommendation to FRA calling upon it to require cab signals and automatic train control on main-line trains in 1971 and for "positive train separation" (PTS) beginning in 1987, and subsequently updated its recommendations in light of changing technology.

Starting in 1994, FRA delivered to Congress its first report on PTC entitled "Railroad Communications and Train Control" (FRA, 1994) The report introduced the term "Positive Train Control" (PTC) for the first time and used it to refer to technology that could automatically intervene to prevent train collisions, control a train's speed, and ensure that trains operate within authorized limits (e.g., stopping short of any work zone). The report also forecasted substantial benefits of advanced train control technology in supporting a variety of business and safety purposes, but noted that an immediate regulatory mandate for PTC could not be justified based upon normal cost-benefit principles relying on direct safety benefits.

In September 1997, FRA asked the newly formed Railroad Safety Advisory Committee (RSAC) to establish a PTC Working Group. The RSAC, an advisory group to the FRA Administrator consisting of representatives from rail labor, rail management, the rail supplier community, and the public reaffirmed the 1994 reports three core objectives for PTC systems were to prevent train collisions, to enforce speed restrictions, and to provide protection for roadway workers. In 1999, the RSAC delivered to FRA's Administrator a report entitled, "Implementation of Positive Train Control Systems" (RSAC, 1999) that attempted to describe a realistic blueprint for successful deployment of the technology. It identified specific actions that government and industry would both need to undertake if PTC was to be deployed successfully across the nation's rail network.

These reports were subsequently followed by a 2003 request by the Appropriations Committees of Congress that FRA update the costs and benefits for the deployment of PTC and related systems. That report, Benefits and Costs of Positive Train Control (Report in Response to Committees on Appropriations, August 2004) ("2004 Report") (FRA, 2004), indicated that under one set of highly controversial assumptions, substantial public benefits would likely flow from the installation of PTC systems on the railroad system. While many of the findings of the 2004 Report were disputed, there were no data submitted to challenge earlier conclusions that the safety benefits of PTC systems were relatively small in comparison to the large capital and maintenance costs.

PTC was on the NTSB's Most Wanted List of Transportation Safety Improvements since the list's inception in 1990. Prior to Chatsworth, the NTSB had repeatedly recommended to the FRA to mandate the installation of PTC. FRA repeatedly declined to take such an action on the grounds that an immediate regulatory mandate for widespread PTC implementation could not be justified based upon traditional cost-benefit principles relying on direct railroad safety benefits. Although FRA fully recognized the benefits of PTC, the significantly adverse cost benefit ratio (on the order of 20:1 against) and the magnitude of the costs (on the order of $9 to $13 billion) precluded a unilateral FRA regulatory mandate for installation. Such an action would represent the most significant change in US train control systems, and the greatest technical and logistic challenge to the US railroad carriers, since the Interstate Commerce Commission (ICC) first ordered the installation of automatic train control systems on US railroads in 1922.

Figure 2. Organizational Regulatory Responsibilities



As is the case with other regulatory bodies in the US, regulations developed by the FRA are reviewed by the Office of Management and Budget (OMB), who considers the impact of the regulation in terms of its cost and efficiency. Proposed regulations must be shown to OMB that they do not create undue financial burdens as well as being supported in terms of the benefits received vice the costs incurred. By both measures PTC failed to satisfactorily meet these tests. Neither could the NTSB effect such a change. NTSB is strictly an investigative agency, which does not have regulatory authority. As a consequence, NTSB did not have any legal authority to issue a regulation mandating the installation of PTC. Even if NTSB had regulatory authority, they would have faced the same OMB evaluation.

Despite the economic infeasibility of PTC based on safety benefits alone, FRA continued with regulatory and other efforts to facilitate and encourage the voluntary installation of PTC systems (see Figure 2). The regulatory reform effort culminated when FRA issued a final rule on March 7, 2005, establishing a technology neutral safety-based performance standard for processor-based signal and train control systems. This regulation, codified as subpart H to part 236 of the Code of Federal Regulations, (GPO, 2005) was carefully crafted to encourage the voluntary implementation and operation of processor-based signal and train control systems without impairing technological development that would make PTC system-wide deployment more economically viable.

Rail Safety Improvement Act of 2008

The accident at Chatsworth, CA changed these dynamics. Prior to the accidents in Graniteville and Chatsworth, the railroads' slow incremental deployment of PTC technologies—while not uniformly agreed upon by the railroads, FRA, and NTSB—was generally deemed acceptable in view of the tremendous costs involved. The U.S. Congress during this same time period had begun

debate on the public policy implications of PTC deployment working on legislation first introduced in May 2007 to effect changes in the federal rail law. Immediately following Chatsworth, however, the legislation effort took on new importance. Just two months after the accident, Congress passed, and then President Bush signed into law, the Rail Safety Improvement Act of 2008 (Public Law 110-432). The Rail Safety Improvement Act of 2008 (also known as RSIA 2008) provided for the most sweeping overhaul of federal rail safety statutes since the Federal Rail Safety Act of 1970. The RSIA 2008 addresses a wide variety of safety related issues ranging from railroad employee's hours of service to implementation of risk reduction programs and technology insertion. When viewed in the context of train control systems RSIA 2008's salient feature, codified at 49 United States Code (USC) 20157, is the requirement for mandatory installation of PTC systems.

The RSIA 2008 statutory requirements for PTC are very specific. PTC systems must be certified by the FRA to provide interoperable positive train separation, over speed protection, protection of roadway workers operating within the limits of their authority, and movement of trains through improperly aligned switches. It must be installed on any railroad line segments over which any passenger or commuter service is provided, as well as any mainline freight segments of the Class 1 railroads that haul Poison by Inhalation (PIH) / Toxic by Inhalation (TIH) materials. In the RSIA 2008, Congress established very aggressive dates for PTC system build-out completion, requiring each subject railroad to submit to FRA by April 16, 2010, a PTC Implementation Plan (PTCIP) indicating where and how it intends to install interoperable PTC systems by December 31, 2015 in priority order based on the level of risk associated with a train accident. Once approved by FRA, railroads would be expected to implement PTC according to that plan. FRA must then annually review the status of railroad compliance with their plan and report railroad status on meeting the schedules in those plans to Congress by the end of 2012.

Implementing RSIA 2008 Requirements

As is the case with most public laws, the RSIA 2008 statutes provides specific outcomes desired by Congress, but does not specify the details for obtaining them. To accomplish these goals, regulatory agencies of the US government are responsible for developing implementing regulations. FRA had previous promulgated regulations for voluntary installations of PTC effective in June 2005 known 49 CFR 236 Subpart H. These regulations provided a set of technology-neutral and performance-based rules for software-controlled microprocessor based train control system designs to replace the existing prescriptive rules for electromechanical and fixed electrical and train control circuitry. While facilitating the development of PTC and the BNSF ETMS system, they did not reflect the RSIA PTC completion mandate of December 31, 2015, or the certification requirement of functionality and interoperability. Consequently FRA undertook development of new regulations, known as 49 CFR 236 Subpart, to address the new statutory requirements of the RSIA.

The first step in developing new federal regulations in the US to implement is to prepare a Notice of Proposed Rulemaking (NPRM) (see Figure 3). The NPRM is official notification by an agency of the intent to implement new regulations. It provides the legal authority for the regulation, the proposed regulatory text, the reasoning behind the regulatory text, and an estimate of the impact on the regulated entities and the public. Any interested party is then invited to offer their suggestions and their concerns regarding the proposal in the NPRM. As part of the process of developing a new NPRM regarding rail safety regulations, FRA attempts to elicit the support of the Rail Safety Advisory Committee (RSAC). The RSAC is a group of rail carriers, labor, vendors, and public

Figure 3. US Regulatory Development Process



users, who are charted to provide advice and recommendations to the FRA Administrator on issues associated with rail safety. FRA asks the RSAC to convene a working group of industry experts to study a problem, determine the issues and provide recommendations as to the most appropriate way to address it. While FRA is not bound by the findings of the RSAC, it attempts to use the results of their efforts to craft regulations to the greatest extent practical. FRA requested, and the RSAC agreed to convene a working group to provide recommendations on the best way in which to implement the statutory requirements of RSIA. The RSAC provided FRA a number of consensus recommendations that were incorporated in the NPRM.

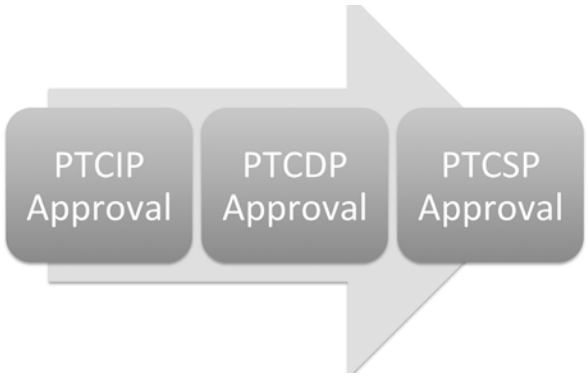
After the NPRM public comment period closed, FRA made modifications to the proposed regulation to address the public comments that were received. The RSAC was requested to review the modifications and provide additional comments. After review of the modified language, the working group agreed with the proposed changes and general framework of the regulation. FRA then prepared a final rule based on the consensus advice and comment, while making decisions on its own on the most appropriate way to address issues where there were no consensus recommendations.

In the case of Subpart I there were three major issues on which consensus could not be obtained. These were the creation of a de minimums exception for the number of PIH cars, the base year that would used for determining what routes would require the installation of PTC, and the requirement for accessibility to the PTC console information by all members of the train control in the locomotive cab. The RSAC members proposed mutually exclusive approaches to resolving these issues proposed, necessitating FRA to exercise its regulatory discretion and craft appropriate regulatory language which was consistent with public safety. The solutions FRA adopted for these were strongly objected to by various stakeholders, who then elected to exercise their rights to obtain redress through the federal court system. While these issues are important are still being addressed by the courts, they are, secondary to the broader general regulatory framework which was agreed upon, and will not be discussed further.

The Subpart I Regulatory Content

Like the earlier Subpart H, Subpart I adopted a performance based approach to meeting the statutory requirements, only tailored to better address the mandatory nature of the RSIA 2008. Subpart I was specifically intended to create a risk based regulatory environment that simplifies paperwork requirements associated with an individual railroad compliance and simultaneously maximizing information sharing and reuse. The major differences previous regulations are associated with the system performance requirements, system certification, system exemptions, and supplier notifications.

Figure 4. Subpart I Certification Process



The performance requirements associated with Subpart I are tailored to address the various the risks and capabilities associated with the PTC system types, specifically non-Vital Overlays, Vital Overlays, Stand Alone, and Mixed. All share common performance requirement that each system type must implement the four PTC function discussed earlier and do so in a reliable and interoperable manner. They differ however in the level of safety each must demonstrate.

Non-Vital overlays by definition are not designed to be failsafe. The risk analysis for these systems must demonstrate that they provide at least an 80% reduction in risk associated with PTC preventable accidents. Vital overlays are designed and built as fail safe systems. The risk analysis for these systems must demonstrate they are fail-safe, but there is no requirement for an 80% reduction in PTC preventable accidents. Stand Alone system design and construction allows for the substitution of the PTC system, and their risk analysis is correspondingly more detailed. These systems must not only be built to a fail-safe standard, but they must demonstrate that they introduce no new unmitigated risks. Mixed systems combine attributes of non-vital, vital, and stand alone systems. Because of their complexity, FRA determines the manner in which compliance is demonstrated on a case-by-case basis in consultation with the railroad.

While not a separate PTC system category, special performance requirements have been established for high-speed rail operations. These tiered performance thresholds are intended to increase safety performance targets as the maximum speed limits increase to compensate for the higher potential adverse consequences of a collision or derailment. These requirements are cumulative as speeds increase.

RSIA 2008 introduced the requirement that systems be certified to be deployed; this is significantly different than the pre RSIA 2008 situation. Before RSIA 2008, and subpart I, FRA approved the use of PTC systems, but to the lesser standard that the system being implemented was at least as safe as the system which it replaced. Post RSIA 2008, and with Subpart I, FRA not only had to approve the use of a PTC system as safe but formally certify statutory compliance. RSIA 2008 certification, as implemented in Subpart I is the final step in a three-step process consisting of review and approval of the PTCIP, PTC Development Plan (PTCDP), and the PTC Safety Plan (PTCSP) (see Figure 4).

Implementation, Development, and Safety Plans

The PTCIP is the regulatory specified document, submitted to FRA for review and approval on or be-

fore 16 April 2010 that provides a risk-prioritized plan for implementing PTC. It provides the specific details on how interoperability is implemented, the railroads with which interoperability has been obtained, the railroads with which interoperability will not be obtained, and copies of any relevant interoperability agreements between the railroads. PTCIP's must be prepared and submitted by all railroads required to implement PTC. By statute that means Class 1 railroads lines that carry PIH/TIH on their line segments with more than 5 million gross tons annually and all railroads, regardless of Class, over which passenger/commuter carriage service is provided. In the situation where there is a host tenant relationship all railroads regardless of class must file a joint PTCIP, though when a Class 1 railroad is the host railroad required to implement PTC, the tenant passenger/commuter service may simply be able to associate itself with the Class 1 PTCIP. The regulations also requires the railroads identify those tracks that they consider to be mainline, and those tracks that they wish to have excluded from PTC installation in the PTCIP. As the main purpose of the PTCIP is to document the deployment plan, the PTCIP will essentially become a historical document when the railroad has completed its PTC implementation, and need only be kept current until installation is complete.

The PTCIP risk prioritization must not only indicate how the PTC system is being implemented from areas of higher risk to lower risk, but must also include the sequence that PTC will be installed on individual line segments, and the basis for this sequencing. There are a number of different risk factors that a railroad may consider when prioritizing its PTC installation, but these risk factors must include segment traffic characteristics, segment method of operations, and route attributes at a minimum. Also required to be included in the PTCIP is a discussion of the schedule by which rolling stock (locomotives) will be equipped with PTC and the numbers of wayside devices required by each line segment. Both of the later provide an indication of a railroad's readiness to actually begin PTC operations.

There is the expectation in the RSIA 2008 statute that Class 1 railroads will also address in the PTCIP how they will implement PTC on all lines that would be covered by the requirements of 49 USC 20156 (Railroad Safety Risk Reduction Program & Technology Implementation Plan). There is, however, no requirement for the Class 1 railroads to install PTC on any of these lines they identified under the provisions of 49 USC 21057 and the Subpart I regulations are silent on this subject.

FRA has elected to use it limited discretionary authority to exempt some segments of track from the requirement to install PTC by the submittal of a main line track exclusion addendum (MTEA) to any PTCIP. FRA will consider exemption requests for installation of PTC for passenger and commuter railroads, or freight railroads operating jointly with passenger or commuter railroads on segments over which limited or no freight railroad operations occur. These are:

1. In passenger terminals where there is no freight traffic and the passenger trains operate at slow speeds.
2. Where there is limited volume of passenger trains operating in the presence of a limited volume of freight.
3. Where the passenger and freight traffic is temporally separated. Temporal separation means only passenger operations, or only freight operations may occur at any time.

Only a passenger railroad may file an MTEA as part of its PTCIP. This may include a PTCIP jointly filed by freight and passenger railroads. In fact, FRA expects that, in the case of joint operations, only one MTEA should be agreed upon and submitted by the railroads filing the PTCIP. Each MTEA must clearly identify and define the physical boundaries, use, and characterization of the track for which exclusion is requested and justify

the exemption request. FRA is not extending any relief or exception to tracks within yards or terminals shared by freight and passenger operations. The collision of a passenger train with a freight consist is typically a more severe condition because of the greater mass of the freight equipment.

FRA also says that a de minims exception to the statutory mandate requiring the installation of PTC systems on any and all main lines transporting any quantity of PIH hazardous materials should also be provided to low density main lines with minimal safety hazards that carry a truly minimal quantity of PIH hazardous materials. Requests for de minims exemptions in the case of PIH/TIH lines must also be included in the PTCIP.

There are also limited exemptions for the operation of non-PTC equipped Class II and III freight railroads operating on PTC territory owned by Class I railroads. Exemptions in this category must be documented in the PTCIP for FRA approval and if granted, expire Dec 31,2020.

The PTCDP forms the basis for the issuance of a Type Approval (TA). The TA is a formal recognition by FRA that the product, if designed and built as described in the PTCDP, will satisfy the statutory requirements. If FRA issues a TA for a PTC system, and other railroads wish to the use the same system as described in the PTCDP associated with the TA, they may do so by citing the TA. This avoids the need to prepare and resubmit duplicative PTCDP documentation. Once issued by FRA, the TA is valid for the longer of either 5 years, or for the duration of time the product associated with the TA is built.

Minor modifications to the system being built under a TA may be proposed by a railroad desiring to utilize a type approved product, but wish to customize the product to a limited extent. In this situation the railroad provides FRA the desired product's TA, along with a detailed discussion of the changes sought. FRA will evaluate the submission. Deepening upon the scope of the changes, FRA may approve the modified system, or require submission of a complete new PTCDP of the proposed system. If the FRA accepts the new PTCDP, FRA may assign a new TA for the modified product.

The submission of the PTCDP or TA and PTCIP is required simultaneously. FRA says that without a clear understanding of the technology involved, a railroad cannot submit a meaningful PTCIP. Knowledge of the capabilities and limitations of a product, and the environment in which it is intended to operate, are essential to develop realistic schedules and determine the interoperability of the system. There are circumstances that may arise for a number of reasons that may preclude submission of a full PTCDP with the PTCIP. A railroad may have an understanding of the functional requirements that they wish a PTC system to meet, but have not been able to establish a source for the desired product. This situation is particularly prevalent among publically funded passenger/commuter railroads. These organizations are often constrained by contracting rules that require competitive bids, precluding identification of the providing vendor.

In situations that preclude submission of a PTCDP or a TA, a railroad may submit a Notice of Product Intent (NPI). The NPI contains a subset of the information required by a PTCDP. While the scope of the information provided in an NPI is insufficient to issue a Type Approval, it does provide FRA sufficient data to make a provisional determination of the products ability to provide the mandatory PTC functionality. Where an NPI has been submitted with the PTCIP and provisional approval has been granted, a railroad has nine months in which to complete submission of a complete PTCDP or identify a previously type approved product. Failure to do so results in disapproval of the PTCIP, and possible civil sanctions.

Where the PTCDP addresses railroad independent product issues, the PTCSP addresses railroad specific product and operation issues. When coupled with the PTCDP and PTCIP, the PTCSP provide a complete picture of a railroad's PTC implementation. There may be differences

from the design and as built condition of the PTC system. The PTCSP provides the railroad a means to identify these implementation related changes that may have occurred, why they occurred, the impact on safety, and any mitigation efforts taken to correct for changes that may resulted in a decrease in the level of safety proposed in the PTCDP or TA. A railroad may, at any time, elect to submit an integrated PTCSP in lieu of a separate PTCDP and PTCSP. The integrated PTCSP, however, must contain the same information as would be found in the individual PTCDP and PTCSP.

The remaining components of the PTCSP are related to the operation, maintenance and test of the railroads installed PTC system. Of these the three most significant are the hazard log, the Operations and Maintenance Manual, and the Training Plan. The hazard log defines all expected hazards associated with the PTC system and their expected frequency. Monitoring of occurrence of these hazards allows the railroad and FRA to determine the efficacy of the required PTCSP risk analysis. A difference in the predicted and actual frequency of hazard occurrence provides tangible evidence that the risk analysis may be incomplete or incorporate invalid assumptions. The Operations and Maintenance (O&M) captures all the required plans, records, processes and procedures for the installation, identification, maintenance, repair, modification, inspection, configuration management and test of the hardware and software components of the PTC system and spares. Proof/Justification of the each element of the O&M Manual is required to be submitted as a part of the PTCSP.

The third item is the training plan. The training plan must be based on task analysis of the required competencies for each of the following categories of individuals:

1. Any individual who installs and maintains elements of the PTC systems,
2. Any people who dispatch and control trains with interaction to the PTC system,
3. Any train crewmember operating PTC equipped trains,
4. Any roadway workers who may depend on PTC for work zone protection, and
5. Direct supervisors of the preceding.

Where applicable, the training must be integrated into other training plans required by regulation (for example into the locomotive engineer training required by 49 CFR Part 240).

Exemptions & Failures

ACSES, ITCS, and ETMS are operational PTC systems with proven service history that predate the passage of RSIA 2008. These systems have shown that they can provide the four PTC core functions and

- Have been previously qualified under 49 CFR 236 Subpart H, or
- Have been previously approved by FRA, has at least 5 years operating history and has undergone an independent verification and validation assessment, or
- Were previously ordered installed by FRA.

While these systems are not exempt from the RSIA 2008 certification requirement, their previous service history systems may qualify them for expedited certification. A system undergoing expedited certification does not require a complete PTCDP or PTCSP. FRA will review any relevant safety case and service history documentation and will make a certification determination on this information. The only significant restrictions on the documentation are if the original documentation is not in English the expedited certification petitioner must provide a certified translation of the document. Expedited certification does not exempt the submission of a PTCIP.

Office Automation Systems function's that are not safety critical are exempt from com-

pliance with Subpart I. A similar provision also applies to locomotive control systems. Locomotive control systems whose functions are not comingled with PTC functionality are also exempt from compliance with Subpart I. Changes in functions that adversely impact the safety performance of the PTC system nullify the exemption.

Subpart I also provide clarification and formalize actions required of both suppliers and railroads when dealing with a system failure that adversely impacts the operation of a PTC system. Railroads discovering a safety issue must notify the supplier, who in turn must notify all other railroads using the product and provide any recommend mitigations. If the supplier discovers an issue, they must notify all railroads using the product and any supplier recommend mitigations. Any railroad receiving notification of an issue, regardless if they have experienced the issue, must behave if the issue occurred on their system, and take the necessary steps to mitigate the problem with out undue delay.

Subpart I differ from previous regulations in that it requires suppliers to notify FRA of system safety issues when the supplier becomes aware of them. FRA already receives notification from the railroad in situations where the railroad discovers a safety issue, but prior to Subpart I notification of supplier identified problems or recommended mitigations was strictly on a voluntary basis by the supplier. In Subpart I, FRA has exercised its general authority and made such notification mandatory from suppliers.

FRA may also grant temporary approval for the irregular diversion of PTC equipped trains over non-PTC equipped track in unusual circumstances. These circumstances might include significant track maintenance work on the PTC equipped route, or "Acts of God" that render the PTC equipped route unusable.

WAY FORWARD TO COMPLETION

All railroads covered by the RSIA 2008 have provided plans that result in completion of PTC installation by December 31, 2015. In fact, several railroads have proposed even more aggressive implementation schedules that would complete installation as early as December 31, 2012. Those plans all have several significant obstacles that must be overcome.

The first is the sheer size and scope of the undertaking. While the exact amount of track that must be equipped with PTC is under constant evaluation based on traffic patterns, it is on the order of 70,000 miles, the number of locomotives that must be equipped with PTC technology components may be as high as 25,000. Completing trackside component installation requires the equipage of approximately 50 miles of track per day, every day, between now and December 31, 2015. Likewise, roughly 13 locomotives must be modified daily between now and December 31, 2015 to achieve compliance with the law. This entire effort must be carried out by the railroads without disrupting existing service, which could negatively impact shippers.

Radio spectrum is a precious commodity with high demands from multiple entities across industries and economic sectors, not just railroads. The Federal Communications Commission (FCC) has already licensed much of the radio spectrum to other users, limiting the available choices to the railroads. The freight railroads were able to purchase licenses from some of the current licensee's for limited amounts of spectrum in the 220 MHz range to provide for interoperable PTC communications. The ability of the commuter and passenger railroads to augment the spectrum already acquired by the freight railroads is greatly influenced by their ability to procure additional spectrum from other current spectrum license holders in the same general frequency range. Their success in acquiring the additional spectrum is a matter of both cost and the willingness of exist-

ing license holders to sell. Without the necessary spectrum, interoperable PTC system deployment is seriously compromised. The issue is not just the availability of the required spectrum, but also the availability of properly configured radios that can operate using the required PTC protocols in that spectrum. Radios with the required operating characteristics do not currently exist in deployable production quantities. While small quantities of prototype and preproduction units exist, any significant delays that may impact the start of full-scale production, of delivery once production has begun, create the potential for compromising timely PTC system deployment.

The vast majority of freight and passenger/commuter operators have elected V-ETMS. While the risks associated with the design, implementation and testing of V-ETMS are mitigated to some extent by the reuse of elements of an already existing ETMS, V-ETMS does not yet exist as a fully deployable system. The interface specifications that must be implemented in V-ETMS are still being finalized and successful completion of all integration and acceptance testing of V-ETMS has not yet been accomplished. In view of the fact that V-ETMS is intended to be a failsafe system, and the hardware and software correspondingly more complex than ETMS, there is a significantly higher potential for emergent technical issues that must be resolved before the V-ETMS system is ready for deployment and operation. As the V-ETMS system is not anticipated to be ready for field integration and acceptance testing until late 2011 or early 2012, significant delays resulting from difficulties that arise in the development and test program would adversely impact successful PTC implementation.

In terms of traditional engineering cost-benefit analysis, the ratio of system implementation costs to the safety benefits realized is in the order of 20:1 against. This creates significant financial pressures on the railroads to try and minimize the scope of PTC deployment. Both the railroads and FRA share a common goal of protecting the safety of the general public and their employees. However FRA, as the agency charged with developing and enforcing the regulations for implementing the RSIA 2008 statutory requirements that were established by Congress as a matter of public policy, has a uniquely different perspective on the best means of meeting that goal than the railroads. The railroads have a duty to their shareholders to maximize their return on investment. FRA's task is to assure safety with reference to, not in deference of the costs associated with that undertaking.

Both privately and publically owned railroads may encounter difficulties with simultaneously raising the necessary capital to support PTC system installation without adversely impacting other costly safety initiatives that address non-PTC related accident prevention efforts. The degree of difficulty in attracting investor capital by privately held freight railroads will undoubtedly be significantly less than publically owned and operated passenger and commuter services. Even in 2009, when revenues were down an average of 22% from 2008, the UP, BNSF, NS, and CSX railroads had total profits exceeding $5.8 billion dollars, and an average total return to investors of almost 35%. Contrast this to the situation of passenger and commuter services that are owned and operated by state and local governments who are significantly more constrained than privately owned railroads in their ability to raise capital. The costs of implementing PTC could potentially affect the extent to which they can provide service, absent additional financial assistance from the government. When the $2.1665 million in collected fares is compared against $7,058 million operating and capital expenses of public passenger/commuter railroads, federal, state, and local government funds are already accounting for over 50% of the total cost of the providing rail service.

As part of RSIA, a highly limited amount of federal funding was made available to support resolution of PTC related design, implementa-

tion, test, and deployment issues. The Rail Road Safety Technology Grant Program was authorized for $50 million per year between 2009 and 2013 to facilitate programs for the deployment of train control technologies, train control component technologies, processor-based technologies, electronically controlled pneumatic brakes, rail integrity inspection systems, rail integrity warning systems, switch position indicators and monitors, remote control power switch technologies, track integrity circuit technologies, and other new or novel railroad safety technologies. This funding, if appropriated for the entire period for which it is authorized and dedicated exclusively to PTC would cover only roughly 3% of the estimated implementation costs. For the fiscal year 2010, FRA received applications requesting over $238 million for eligible projects for which only $50 million was available.

Although the obstacles to successful PTC implementation by the end of December 2015 are serious, it is premature to project failure. Railroads have already spent significant amounts of their infrastructure capital to work on voluntary implementation of PTC before the statutory mandate was enacted and capital investments that are directly transferable to addressing the technical challenges associated with the RSIA 2008 PTC mandate. The railroads are also working with a previously unprecedented level of effort to collaboratively find cost effective solutions to the technical challenges. While there are some disagreements between the FRA and the railroads regarding the best resolution of some of the challenges enumerated here, all parties appear fully committed to PTC deployment.

NOTE

The views and opinions expressed herein are those of the authors and do not necessarily state or reflect those of the United States Government, the Department of Transportation, or the Federal Railroad Administration, and shall not be used for advertising or product endorsement purposes.

REFERENCES

American Public Transportation Association (APTA). (2010). *Public Transportation Fact Book 2010, Appendix B: Transit Agency and Urbanized Area Operating Statistics*. Washington, DC: APTA.

Association of American Railroads (AAR). (2005) *Manual of Standards and Recommended Practices, Section K- Railway Electronics* AAR Publications.

Association of American Railroads (AAR) Policy and Economics Department. (2010). *US Freight Railroad Statistics, November 2010*. AAR Publications.

European Commission (EC) (2010) *2010/79/EC Commission Decision of 19 October 2009 amending Decisions 2006/679/EC and 2006/860/EC as regards technical specifications for interoperability relating to subsystems of the trans-European conventional and high-speed rail systems* (notified under document C(2009) 7787)

European Union (EU) Eurostat. (2010) Regional *Transport Statistics- Railway Transport Measurement* - retrieved from http://epp.eurostat.ec.europa.eu/portal/page/portal/transport/introduction

Federal Railroad Administration (FRA). (1994). *Railroad Communications and Train Control*. FRA.

Federal Railroad Administration (FRA) (2004) *Benefits and Costs of Positive Train Control. Report in Response to Request of Appropriations Committees. August 2004*

Federal Railroad Administration (FRA). (2005). *FRA Emergency Order Number 24- Emergency Order Requiring Special Handling, Instruction, and Testing of Railroad Operating Rules Pertaining to Hand Operated Main Track Switches*. FRA.

Federal Railroad Administration (FRA) (2006) *U.S. DOT/FRA - Letter Approving BNSF's Product Safety Plan Ver. 2.1 dated 21 December, 2006 Docket ID: FRA-2006-23687*

Federal Railroad Administration (FRA) Office of Railroad Development. (2009). *North American Joint Positive Train Control (NAJPTC)*. Project. Research Results.

Federal Railroad Administration (FRA) Office of Safety Analysis. (2010), *Operational Data Tables, Table 1.02* retrieved from http://safetydata.fra.dot.gov/OfficeofSafety

International Monetary Fund (IMF). (2010). World Economic Outlook. *Database*, *2010*, n.d. retrieved from http://www.imf.org/external/pubs/ft/weo/2010/02/weodata/index.aspx

International Union of Railways (IUC). (2010) *On-line Statistics* retrieved from http://www.uic.org/spip.php?article593

National Transportation Safety Board (NTSB) (1993) *Railroad Accident Report (RAR-94-01): Derailment of Amtrak Train No. 2 on the CSXT Big Bayou Canot Bridge Near Mobile, Alabama September 22, 1993.*

National Transportation Safety Board (NTSB). (2003) *Railroad Accident Report (RAR-03-04): Collision of Burlington Northern Santa Fe Freight Train With Metrolink Passenger Train, Placentia, California, April 23, 2002.*

National Transportation Safety Board (NTSB). (2005) *Railroad Accident Report (RAR-05-04): Collision of Norfolk Southern Freight Train 192 With Standing Norfolk Southern Local Train P22 With Subsequent Hazardous Materials Release at Graniteville, South Carolina, January 6, 2005*

National Transportation Safety Board (NTSB) (2008) *NTSB Most Wanted List Transportation Safety Improvements, 2008-2009*

National Transportation Safety Board (NTSB). (2010) *Railroad Accident Report (RAR-01-01): Collision of Metrolink Train 111 with Union Pacific Train LOF65–12 Chatsworth, California, September 12, 2008*

Railroad Safety Advisory Committee (RSAC) (1999) *Implementation of Positive Train Control Systems*

Strauss, J. F. (1998). *The Burlington Northern An Operational Chronology 1970-1995, Friends of the Burlington Northern Railroad*. Wisconsin: West Bend.

U.S. Department of Transportation (DOT) Bureau of Transportation Statistics. (2010) *National Transportation Statistics 2010* retrieved from http://www.bts.gov/publications/national_transportation_statistics

U.S. Government Printing Office (GPO) (2005) Standards for Development and use of Processor based Signal and train Control Systems, Final Rule, *Federal Register, 70*(43), U.S. Government Printing Office (GPO) (2008) *PUBLIC LAW 110–432—OCT. 16, 2008 FEDERAL RAIL SAFETY IMPROVEMENTS- Rail Safety Improvement Act of 2008*

U.S. Government Printing Office (GPO). (2010). Positive Train Control Systems Final Rule. *Federal Register*, *75*(10).

KEY TERMS AND DEFINITIONS

Association of American Railroads (AAR): Trade group representing primarily the major freight railroads of North America

Federal Communications Commission (FCC): An independent U.S. government agency responsible for regulating interstate and interna-

tional communications by radio, television, wire, satellite and cable.

Federal Railroad Administration (FRA): A modal administration of the U.S. Department of Transportation responsible for promulgating and enforcing rail safety regulations.

Code of Federal Regulations (CFR): An executive branch codification of U.S. Administrative Law that implements legislative branch statutory laws

Interstate Commerce Commission (ICC): A former independent U.S. government agency responsible for supervising transportation carrier rates and safety

National Transportation Safety Board (NTSB): An independent U.S. government agency responsible for accident investigation

Office of Management and Budget (OMB): An independent U.S. government agency responsible for advising the US President on budget and policy matters

Positive Train Control (PTC): Integrated command, control, communications, and information systems for controlling train movements.

Public Policy: A course of action on a particular issue affecting the public, along with the associated statutory law, administrative law, and funding priorities

Rail Safety Improvement Act (RSIA): An amendment to title 49 USC to prevent railroad fatalities, injuries, and hazardous materials releases.

Surface Transportation Board (STB): An independent U.S. government agency responsible for supervising transportation carrier rates. Successor to the ICC

Toxic by Inhalation (TIH): Types of hazardous material that enter the body through the respiratory system that will result in extremely severe injury or death (also known as Poison by Inhalation)

ENDNOTES

1. The STB defines a Class I railroad in the United States as "having annual carrier operating revenues of $250 million or more. It is one of three classes of railroads defined by the STB. The other two are Class II: Carriers having annual carrier operating revenues of less than $250 million but in excess of $20 million and Class III: Carriers having annual carrier operating revenues of $20 million. All revenue values are uncorrected for inflation. In 2009 the inflation adjusted revenue for designation as a Class I Railroads was operating revenue of $378.8 million or more. (Source 49 CFR Part 1201)
2. Revenue passenger miles, an indicator of use, is obtained by multiplying the number of revenue-paying passengers aboard the vehicle by the distance traveled.
3. In railroad signal systems, a block is zone of control in which a train operates. A "fixed block" is predetermined stationary track segment between two points in which a train operates. Trains move from block to block. A "moving block" is the zone of control around a train whose size and location are continuously calculated in real time. Unlike a "fixed block", the zone of control moves with the train.

Chapter 2
The Model-Driven openETCS Paradigm for Secure, Safe and Certifiable Train Control Systems

Jan Peleska
University of Bremen, Germany

Johannes Feuser
University of Bremen, Germany

Anne E. Haxthausen
Technical University of Denmark, Denmark

ABSTRACT

A novel approach to managing development, verification, and validation artifacts for the European Train Control System as open, publicly available items is analyzed and discussed with respect to its implications on system safety, security, and certifiability. After introducing this so-called model-driven openETCS approach, a threat analysis is performed, identifying both safety and security hazards that may be common to all model-based development paradigms for safety-critical railway control systems, or specific to the openETCS approach. In the subsequent sections state-of-the-art methods suitable to counter these threats are reviewed, and novel promising research results are described. These research results comprise domain-specific modeling, model-based code generation in combination with automated object code verification and explicit utilization of virtual machines to ensure containment of security hazards.

INTRODUCTION

In 2009 German Railways initiated a discussion on a novel development paradigm for railway control systems. Motivated by concerns related to development costs, overall system quality, safety and security, they advocated the publication of re-usable control system code as *Free/Libre Open Source Software FLOSS* (Hase, 2009a; Hase, 2009b). The focus of this discussion was on the *European Train Control System ETCS,* because (1) as of today, ETCS is the most challenging European railway technology project, (2) the European Union has formulated explicit requirements

DOI: 10.4018/978-1-4666-1643-1.ch002

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

regarding the introduction of ETCS in European countries, and (3) the public availability of the ETCS standard allows industries and research communities to perform analyses of and contribute to the ETCS concepts and technologies without infringing vendor-specific property rights. As one result of the initial discussions, a comprehensive list of development, verification and validation (V&V) artifacts was identified which should also be published under the FLOSS regime. These artifacts comprised specifications, designs and V&V results, such as correctness proofs, test suites and test results obtained with the publicly available software. Moreover, a well-defined open tool chain suitable for generating executable software code, V&V results and certification credits from these artifacts was advocated (Feuser & Peleska, 2010; Hase, 2011).

In this chapter the authors present a comprehensive work flow definition for developing railway control systems according to the FLOSS paradigm. We consider the following aspects of this exposition as our main contributions:

- It is described how the ETCS standard can be formalized using a modeling approach based on a *domain-specific language (DSL)*. With this formalization at hand, concrete developments may be regarded as refinements and extensions of the abstract model represented by the standard, and conformance of the development to the standard can be established in a more rigorous way.
- The general openETCS paradigm is specialized to a *model-driven* openETCS paradigm, where developments focus on formally modeling the expected system behavior, while source code may be generated from the models in an automatic way.
- The workflow description is associated with a hazard analysis identifying the remaining safety and security threats still present when performing developments according to the model-driven openETCS paradigm.
- Effective methods for countering each hazard are described, so that the resulting development approach will guarantee a higher degree of safety and security and allow for more effective certification than the conventional approaches performed today by most of the railway suppliers.

The full ETCS standard covers train control computers, track elements and functionality to be integrated in track side interlocking systems. Following (Hase, 2011) we focus in this exposition on the train control computer, called the *European Vital Computer EVC* which is responsible for *automated train protection (ATP)*. Observe, however, that the underlying paradigm, methodology and techniques described in this chapter are applicable to any railway control system development where

- A publicly available standard exists specifying system requirements in a generic way,
- A model-driven development and V&V approach is chosen, and,
- The contributing parties are willing to publish development and V&V artifacts according to the FLOSS principle.

BACKGROUND

In the original motivation of the openETCS initiative presented by Hase (2009a, 2009b, 2011), the author relates security issues observed in standard software products directly to threats to be expected in ETCS developments. For example, he argues that if backdoors could be integrated into standard commercial closed source products, the same could happen to railway control systems software. It should be emphasized, however, that

the development and V&V processes for standard software products differ from those applicable to railway control system software in a considerable way: the latter have to follow a rigorous approach specified in the standards (European Committee for Electrotechnical Standardization, 2000, 2001a, 2003, 2001b, 2001c) whose observation is checked by certification authorities. In particular, these standards require independent V&V for the higher safety integrity levels. As a consequence, deliberate integration of malware into, say, an EVC development, would only succeed if

1. Developers and members of the V&V team jointly agreed to perform and "overlook" a malware injection, or,
2. The V&V team would unintentionally fail to detect this in the code.

The probability for the occurrence of situation 1 is low according to the authors' assessment, because it requires two independent parties to agree on a malicious act that would be traceable to the persons involved, in case of an accident occurring due to the activation of the malware. On the other hand, the probability for situation 2 to occur cannot be neglected, due to the growing complexity of railway control systems and the fact that V&V teams in closed source developments may be understaffed in order to reduce costs. Therefore we agree with Hase (2009a, 2009b, 2011) that the public peer review of models and code is highly desirable in order to avoid situations of the type 2 above. On the other hand we presuppose the existence of a trusted V&V team that will not deliberately overlook unintended or malicious injections into the code.

The core documents of the ETCS which are applicable to the European Vital Computer are the *ETCS System Requirements Specification (SRS)* (UNISIG, 2006a, 2006b, 2006c, 2006e, 2006f, 2006g). The domain-specific modeling language introduced in this chapter aims at formalizing these parts of the standard. The underlying functional requirements have been specified in (European Railway Agency, 2007). Our approach to start the formalization already on the level of the ETCS standards is justified by the fact that these standards have been elaborated in the style of functional and structural system specifications, applicable to on-board train control computers, track elements and interlocking systems.

Several known approaches to ETCS formalizations focused on the communication system. Platzer & Quesel (2009) developed a mathematical model for the communication protocols supporting the train separation mechanisms defined by the ETCS system requirements specification (Pachl, 2002; UNISIG, 2006b). This model is used to verify controllability, reactivity, safety and liveness. For the purpose of modeling the behavior of the ETCS communication channels, Trowitzsch & Zimmermann (2006) introduced a graphical formalism which is based on UML and Petri nets. A very first approach to formally modeling the ETCS specification and implementation process was proposed by Hörste & Schnieder (1999). These authors already proposed a model-to-text transformation to generate source code from formal models, but – in contrast to the domain-specific approach advocated in this chapter – they also relied on the utilization of Petri nets which (like UML) is a wide-spectrum formalism. The advantages of domain-specific formalisms for modeling railway control systems have been thoroughly investigated during the last decade, see (Haxthausen & Peleska, 2011; Mewes, 2010) and the references given there.

Our DSL design uses both function blocks processing the signal flow and state machines for discrete control. This design decision has been influenced by Berry (2003), who described the combination of block diagrams and safe state machines for modeling continuous and discrete control flow for safety-critical systems: in complex embedded systems both language elements are

needed, because the control of sensors and actuators typically involves continuous control of the data flow, while the modeling of communication protocols requires state machines describing discrete transitions of the protocol machines involved. Berry's suggestions have been integrated in the SCADE-tool of Esterel Technologies, which also provides full code generation for safety-critical systems, but, again, does not provide domain-specific language elements tailored for ETCS.

The model-driven openETCS paradigm relies on a publicly available *platform independent model (PIM)* serving as a "starting point" to *platform specific models (PSMs)* elaborated by suppliers in the course of concrete EVC developments. To fully exploit the benefits of the public peer reviewing process performed for the PIM it is crucial that only controlled transitions from PIMs to PSMs are performed, so that as many original PIM elements are identifiable in the PSM (and, finally, in the software integrated on the target hardware), as possible. Controlled model transformations require formal concepts how source models can be extended, restricted and modified. In the context of the UML2 this has led to the *package merge* concept described in (Object Management Group OMG, 2010b, pp. 130-139). This concept has been analyzed in depth and improved by Dingel, Diskin & Zito (2008).

While the UML2 is based on the MOF meta-metamodel as specified in (Object Management Group, OMG, 2010a), the DSL introduced by the authors uses the *Graph-Object-Property-Relationship-Role (GOPRR)* meta-metamodel (Kelly, & Tolvanenm, 2008, pp. 411-414). The ETCS DSL metamodel and the PIM have been developed by the authors using the MetaEdit+ tool which supports the development of DSLs on the basis of GOPRR. Model extension and modification is supported by means of import and export mechanisms, and through a *copy-by-reference* function allowing to re-use model components in different contexts (Kelly, & Tolvanenm, 2008, pp. 394-407). Mewes (2010) presents a thorough discussion of the OMG and GOPRR meta-metamodels in the context of railway control systems modeling.

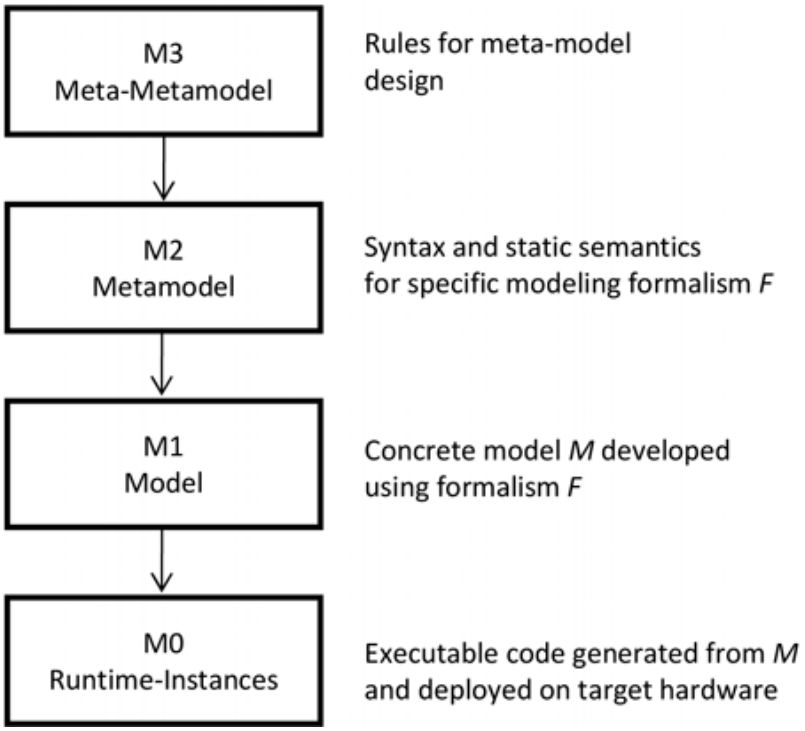
Our specific approach to object code verification presented in Section *Object Code Verification* has been inspired by Pnueli, Shtrichman & Siegel (1998). We have a similar formal basis: for example, our notion of I/O-equivalence is a specialization of the "correct implementation relation'' defined in (Pnueli, Shtrichman & Siegel, 1998). Using this specialization, it has been possible for us to simplify the equivalence proofs in a considerable way.

MODEL-DRIVEN ENGINEERING AND DOMAIN-SPECIFIC LANGUAGES

The general notion of model-driven development considers software development as a transformational approach where code is derived – if possible by completely automated transformations – from models. Models focus on the functional and structural properties of the systems to be developed, while abstracting from programming details that are ideally filled in automatically during transformation phases. To support this approach it will often be necessary to provide several models presenting the system on different levels of abstraction, connected to each other by refinement relations: a *platform-independent model (PIM)* specifies system properties without referring to specific target platforms where the software to be developed might be deployed on. In contrast to this, a *platform-specific model (PSM)* is enhanced by all the properties depending on a concrete HW/SW embedding, such as concrete data formats for interfaces, utilization of operating system services and software allocation on processors.

A fully automated transformational approach strongly depends on the semantic precision of the underlying models. To clarify the design process for appropriate modeling languages, the Object

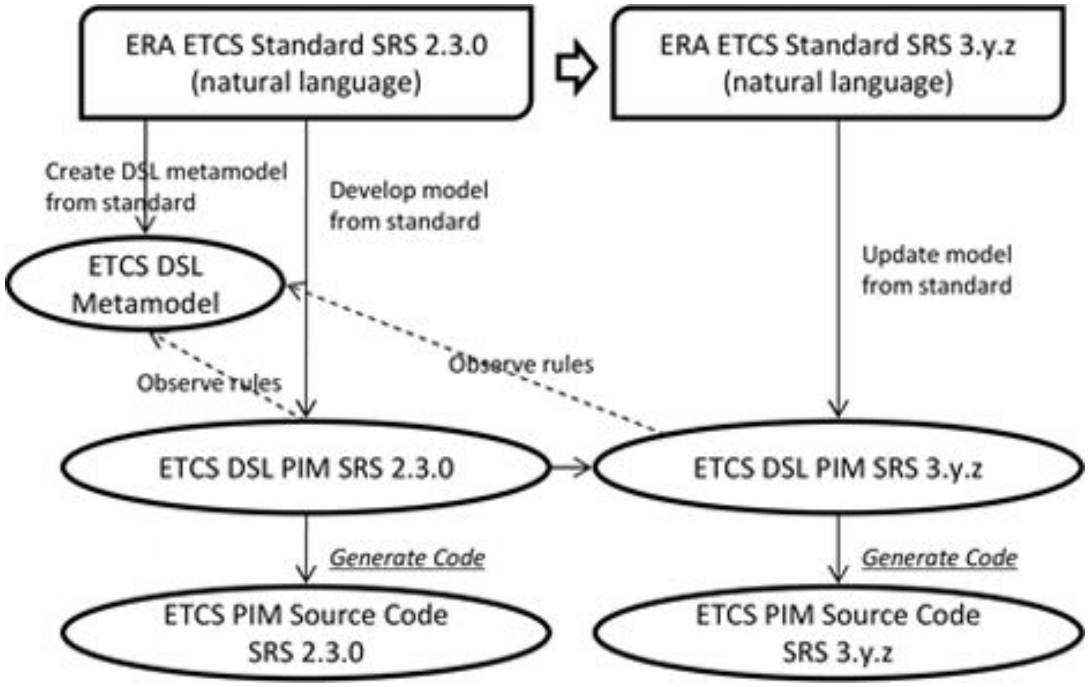
Figure 1. The meta-modeling hierarchy



Management Group, OMG (2010a) has therefore characterized the language design process as sketched in Figure 1. A *meta-metamodel* specifies rules for designing a "universe" of modeling languages. Compared to the design of textual programming languages, the meta-metamodel corresponds to a description of techniques for specifying grammars of such languages, such as BNF or syntax graphs. The *metamodel* specifies the rules for admissible syntax and static semantics of a concrete formalism. Comparing this again to the design of textual languages, this corresponds to defining the concrete grammar of one language (e.g., the BNF production rules for the C programming language). Some metamodels also specify rules for deriving behavioral semantics from syntactically correct models, so that the complete system behavior can be derived from models developed according to these metamodels. A concrete *model* specifies the system to be developed in the formalism defined by its metamodel. If the metamodel does not provide rules how to derive dynamic behavior from syntactic model representations, a *transformational semantics* can be introduced by translating the model into a – typically more concrete – model whose metamodel provides rules for determining the behavioral semantics (*model-to-model* transformation). For PSMs concrete code can be derived by means of so-called *model-to-text* transformations.

The notion of *model-driven engineering (MDE)* emphasizes that the formalisms to be applied for model-driven development should be domain-specific (Kent, 2002; Schmidt, 2006): a *domain-specific language DSL* is based on a metamodel already containing the terms and concepts, as well as the syntactic and semantic restrictions – formally speaking, the *type system* – of the application domain. It is rightly claimed that using an adequate DSL for expressing the

Figure 2. ETCS standard, domain-specific metamodel, platform-independent models and code



properties of a system to be developed improves the communication between domain experts and system or software developers, reduces modeling errors violating the rules of the application domain and facilitates code generation by reducing the range of models to be transformed to those occurring in this specific domain (Kelly, S., & Tolvanenm J.-P., 2008).

The concept of DSLs contrasts to so-called *wide-spectrum languages* that utilize language elements applicable in any application domain. UML 2, for example, certainly is a wide-spectrum formalism. Its metamodel, however, allows specifying so-called *profiles* which may be used to extend the UML by domain-specific language elements. In contrast to this, we will introduce a DSL in Section **'Domain-Specific Modeling for openETCS Developments'** whose metamodel is not based on the UML, but has been elaborated from scratch in a different meta-metamodeling framework.

MODELING, GENERATION, VERIFICATION, VALIDATION AND CERTIFICATION IN THE OPEN ETCS APPROACH

The Model-Driven openETCS Paradigm

The openETCS paradigm specifies a process for ETCS developments. While the initiators of the paradigm already identified open standards, models, source code and proofs (used as a loosely defined term for all artifacts related to V&V) as characteristics of openETCS, we advocate a model-driven approach to openETCS in this chapter, as illustrated in Figures 2, 3 and 4.

As shown in Figure 2, the current version of the ETCS standard is analyzed initially to create a metamodel for a domain specific modeling language. As described in Section *Domain-Specific Modeling for openETCS Developments* we propose to transform the EVC-related portions of

Figure 3. From platform-independent to platform-specific models and associated code

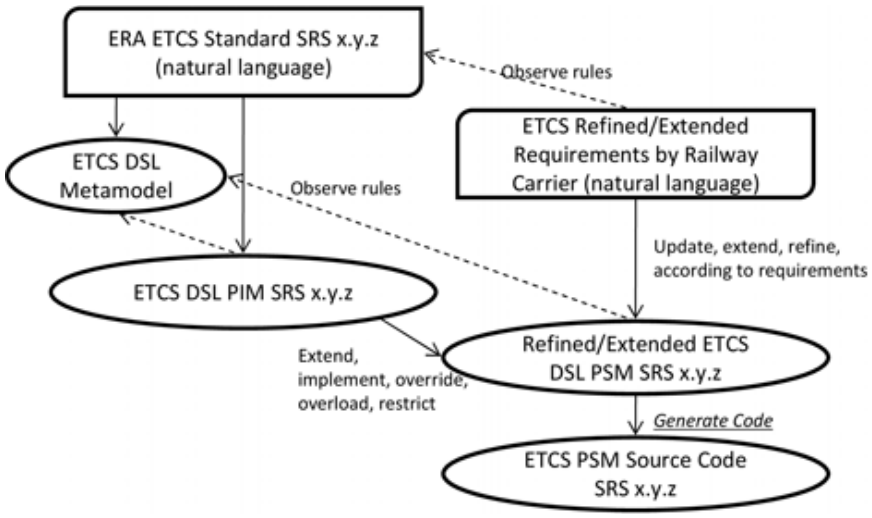
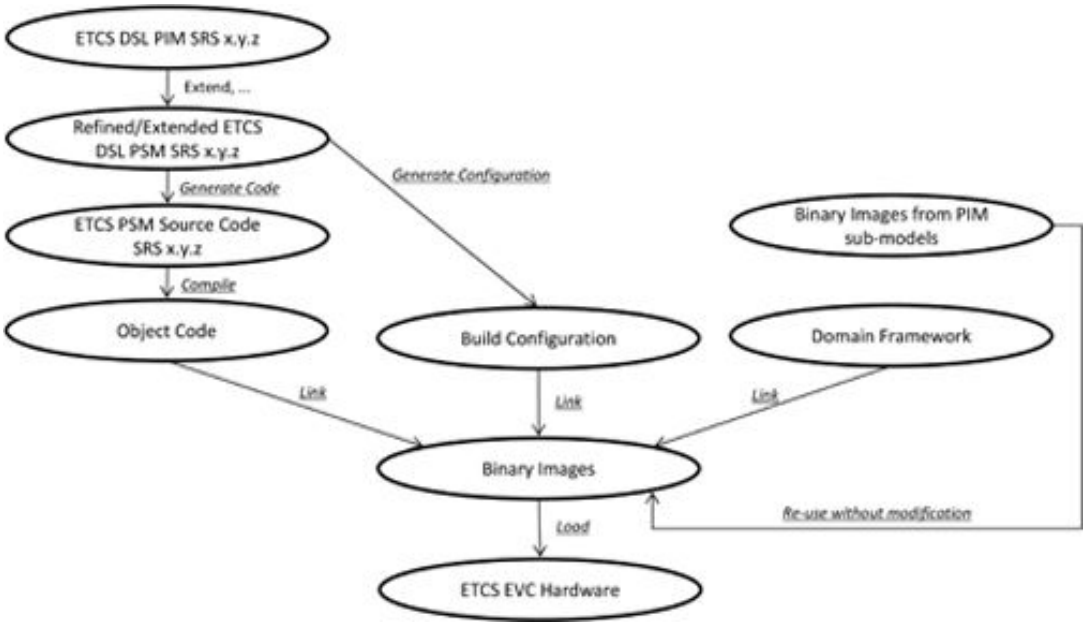


Figure 4. Build process for binary image and loading onto target hardware



the ETCS standard into a platform-independent model. Since the ETCS standard contains a high-level natural language description of the EVC behavior in different operational situations, this approach is justified: the standard specifies behavior as far as this is independent on concrete hardware adaptations or national (i.e. country-specific) functional extensions. As a consequence, the associated PIM covers a level of detail where datagram structures are fully specified down to byte level, and crucial control algorithms are elaborated up to, but excluding, concrete hardware interfaces. The decision to rather use a DSL than a wide-spectrum formalism like UML2 or SysML

is justified in Section *DSL Design Decisions* below.

The metamodel remains unchanged, if a revised version of the ETCS standard is published (SRS 3.y.z in Figure 2): the metamodel covers all language elements which were anticipated to be utilized in the current and future versions of the standard. Therefore only the PIM has to be updated to the standard's new baseline, still observing the same modeling rules as for the initial PIM. Since PIMs already specify structure and behavior, it is possible to generate code from these models. This code will typically contain the fundamental control algorithms specified by the ETCS standard, and it is an explicit openETCS objective that it shall be re-used without change in every concrete EVC development. Putting this objective into practice is facilitated if the PIM or at least the code generator can refer to an application program interface (API) of an operating system. This API is used to specify access to protected resources (e.g. hardware interfaces), different levels of concurrency (creation of threads and processes) and synchronization mechanisms (e.g. semaphores) protecting critical sections. Following the terminology of (Aeronautical Radio Inc., 2005), this API is called the *EVC application executive (APEX)* in the remainder of this chapter. We even advocate providing an operating system implementing the EVC APEX for the PIM, so that portions of the PIM can be transformed into executable binary images. It is explained in Section *Containing Security Issues Implied by Supplier-Specific Closed-Source Code* how these executables may be directly integrated into specific hardware platforms by means of hardware virtualization.

In the model-driven openETCS approach the DSL metamodel and the PIM are made publicly available. For the source code there are two options: (1) the source code is published just as in any other FLOSS development, or (2) a CASE tool suitable for editing PIMs, the source code *and* the code generator are published, so that any member of the openETCS community may experiment with ETCS developments by changing the PIM and generating new code from the modified model. Currently the openETCS community is much in favor of the second option, because this seems to be the most promising way to identify errors and work out fixes and improvements. Observe, however, that the source code has to be published in any case, in order to identify a well-defined code baseline to be compiled and integrated into a concrete EVC target platform.

Figure 3 sketches the transition from the general standard and associated PIM to a specific EVC development and associated source code generation. A concrete development would typically be initiated by a railway carrier or by a system supplier, and we anticipate the following refinements and extensions to the general standard: (1) the abstract interfaces of the PIM are made specific, due to concrete specification of an EVC hardware platform and concrete interfaces to the other on-board systems of ETCS trains, (2) national functional extensions are specified, providing additional functionality not addressed by the ETCS standard, and (3) restrictions to the full functionality of the standard are defined. As a consequence, some portions of the PIM have to be adapted in order to create a platform specific model, while others remain unchanged. The PSM contains references to the PIM sub-models re-used without modifications, so that their binary images can be directly re-used with the help of hardware virtualization. The EVC APEX calls performed by the re-usable PIM sub-models and referring to abstract interfaces are routed to the new PSM components mapping these calls to concrete hardware interface driver invocations.

The PIM is extended in order to capture new functional requirements, some PIM functionality is overridden by new solutions and some PIM portions may be removed because the associated EVC functionality is not to be implemented. At the end of the PSM elaboration this model contains (references to) all applicable information of the PIM plus the adaptations of the concrete development.

From the syntactic point of view the PSM requires the same modeling elements as the PIM, so the same metamodel applies for PSM development. This implies that the same code generator may be applied on the PSM. The generated code, however, may still contain references to operations whose implementations have not been defined within the PSM itself, but in the so-called *domain-framework*, which is addressed in Figure 4.

In Figure 4 the build process for a concrete EVC development is depicted. The source code generated from the PSM is compiled and the resulting object code is linked together with object code from the domain framework which contains platform-specific drivers, operating system, hypervisors, virtual machines and code integrity checkers identifying at run-time whether consistent binary images are executed on the EVC hardware. Moreover, re-usable binary images generated from the PIM are added to the configuration. This process is governed by a build configuration specification derived from the PSM. The binary images stemming from the PIM sub-models and the development-specific images originating from the PSM have to be executed on different virtual machines, in order to cope with different APIs and to support partitioning as explained in Section *Containing Security Issues Implied by Supplier-Specific Closed-Source Code*. These security-related aspects are discussed in Section *"Threat Analysis for Developments Following the openETCS Paradigm"*.

The V&V obligations arising from the development approach described here are discussed in Sections *"Transformational Semantics and Model Validation"* and *"Object Code Verification"*.

Threat Analysis for Developments Following the openETCS Paradigm

A major objective of the openETCS paradigm is to reduce safety- and security-related threats to an EVC development by means of the public peer reviewing process. While it is obvious that this peer review will increase the completeness and consistency of the platform independent model, the novel workflow and the model-based approach require a more thorough threat analysis which is performed in this section.

Threats

As a starting point it is assumed that in principle any artifact of the development lifecycle may be corrupted due to

- Unintended human error,
- Deliberate manipulation by malicious agents, or,
- Tool malfunction.

Analysis of the artifacts captured in Figures 2, 3, 4 indicate that an error in any of these may – with varying degrees of probability – eventually lead to a safety-critical error in the binary images integrated in the EVC hardware. It is therefore the objective of this analysis to investigate where – compared to a conventional closed-source development process – the model-driven openETCS approach promises a higher degree of safety, and where additional measures may be required in order to ensure sufficient trustworthiness for the final integrated HW/SW product.

Effects of the Public Peer Reviewing Process

It is obvious to see that the artifacts ERA ETCS Standard, ETCS DSL Metamodel, PIM, and the re-usable code generated from the PIM will directly benefit from the public peer reviewing process, in particular, if the generation, compilation and V&V tools are also made publicly available. In that case the openETCS community can be relied on detecting a substantial portion of

- Logical errors in standard, metamodel and PIM, as well as
- Programming errors in the code generator, compiler and the V&V tools.

From the point of view of the certification process the public peer reviewing of standard, metamodel, and PIM is just an extension of the group of review participants. For the tools, however, certification rules require *tool qualifications*, and broad utilization of a tool by a large community does not automatically result in qualification credit. Therefore the same tool qualification requirements as for conventional closed source developments apply to the model-driven openETCS approach.

No Qualification of Development Tools

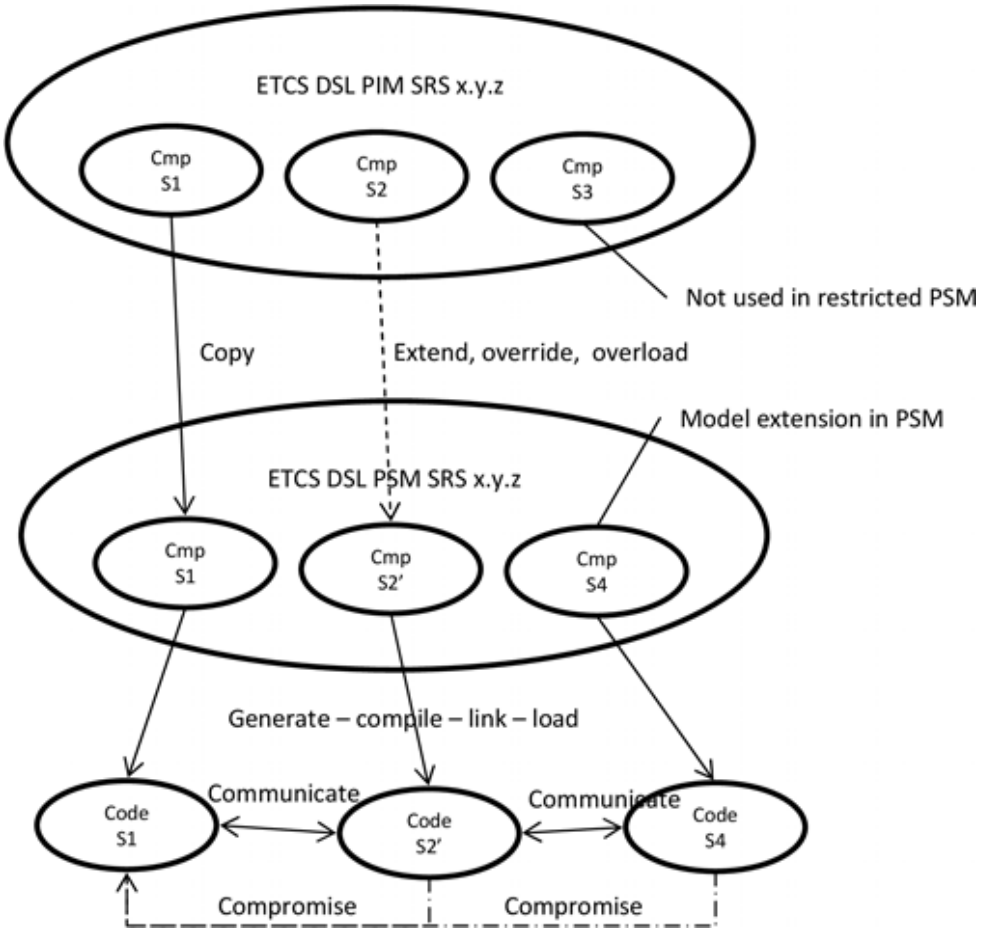
All current standards for safety-critical systems in the railway, avionic or automotive domains distinguish between two classes of software being applied or produced during the development lifecycle: for some software components continuous performance without failure is vital for overall system safety, because erroneous behavior – even if detected immediately after its occurrence – cannot be masked or undone. This characterization typically applies to a subset of on-board control software. In contrast to that, other software components will not threaten system safety if every failure is *revealed.* This applies to fail-stop on-board components whose failure just causes system transitions into stable safe state. Moreover, this characterization applies to all types of development, verification and validation tools: if it can be reliably checked whether, for example, the object code produced by a compiler is correct, than a failure to produce correct code will not impair system safety, because this object code will be discarded and never be deployed on the target hardware.

For model-driven openETCS we therefore advocate an approach where the development tools (model-to-text generator, compiler, linker, loader) do *not* have to be qualified because every potential error will be revealed by a verification tool using *orthogonal methods*. The latter term means that the verification tool does not share any algorithms or code with the development tool whose outputs are to be verified. If this is ensured the probability that a development tool error leading to an erroneous output is masked by a verification tool error overlooking exactly this type of output error can be neglected. As described in Section *Transformational Semantics and Model Validation* the outputs of the *source code generator* → *compiler* tool chain will be verified against the PIM, and further verifications are used on the integrated HW/SW system. In order to justify this approach it has to be understood that the orthogonal object code verification is exhaustive: instead of just testing with some input data whether the object code produced leads to system behavior compliant with PIM and PSM it is formally proven that all possible behaviors resulting from object code execution are compliant with the behaviors admissible according to PIM and PSM.

Threats to PSM and Associated Transformations

Figure 5 illustrates the threats implied by the PSM and its associated transformation artifacts. In principle, PSM extensions to the PIM also threaten the unmodified code originating from the PIM: apart from the intended communications, the code components resulting from PSM extensions and modifications may pass corrupted data over legal and *covert channels* (for example, shared memory sections) to compromise other code components at runtime. In Section *Containing Security Issues Implied by Supplier-Specific Closed-Source Code* we suggest the introduction of hardware virtualization to counter these threats.

Figure 5. Threats implied by PSM extensions to PIM and supplier-specific implementation



Domain-Specific Modeling for openETCS Developments

DSL Design Decisions

In this section we introduce a domain-specific formalism for EVC developments. The DSL is suitable for formalizing the EVC-related portions of the ETCS standard itself, so that concrete developments may be considered as specializations or refinements thereof. The selection of a domain specific (modeling) language (DSL) is motivated by the objective to directly use the terms and concepts introduced by the standard in the models. This facilitates the mapping from the natural-language text of the standard to the formalized model in a considerable way, so that the model verification against the original documents will be easier in comparison with the alternative, where the terms and definitions of the standard have to be formalized using the elements of a wide-spectrum language such as UML2 or SysML. Summarizing the requirements for a suitable ETCS DSL, we need a modeling formalism that

- Captures the terms, definitions and concepts of the current ETCS standard, as far as applicable to the EVC,

- Can be re-used for all future extensions or modifications of the ETCS standard we are able to anticipate today,
- Covers all modeling aspects from the ETCS standard, via system requirements and design, down to algorithmic specifications and HW/SW integration, and,
- Provides concepts for model extension, restriction and controlled modification, in order to facilitate the transformation from platform independent to platform specific models.

An analysis of the EVC-related documents (UNISIG, 2006b, 2006c, 2006e, 2006g) of the standard reveals the following characteristics which are specific to the ETCS domain:

- The overall EVC behavior is governed by the so-called *transitions table* (UNISIG, 2006e) specifying the operational *modes* of an EVC and the *functions* which are active in each mode, as well as the *transitions* between modes with their guard conditions and priorities.
- The availability of modes and associated functions, as well as mode transitions, is dynamically controlled by *(application) levels* depending on the available track side equipment and the resulting interaction possibilities between EVC, track elements and radio block centres (i.e., interlocking systems).
- All data items exchanged between EVC and track side equipment and some general state variables are fully determined by the *ETCS language* (UNISIG, 2006c) defining the structure of admissible communication telegrams, as well as identifiers and types of the data items they contain.

These observations strongly motivate to incorporate the concepts of transition table, levels and ETCS language directly into the DSL. In particular, as explained in Section *Model-Driven Engineering and Domain-Specific Languages* above, the DSL type system allows us to restrict general concepts in a way such that only models can be built which are *a priori* compliant with the standard. Consider, for example, the transitions table which is discussed in more detail below: while a wide-spectrum formalism will also offer syntactic elements to specify state transition systems, the encoding of the transitions table in the DSL metamodel ensures that only the control modes that are defined in the ETCS standard are addressed in the table. Similarly, wide-spectrum languages will provide notation to specify hardware interfaces. In contrast to this, the specific hardware interfaces pre-defined in the standard – for example, interfaces between EVC and odometer or between EVC and Eurobalises – are also encoded in the DSL metamodel, so that only admissible hardware interfaces can be applied during modeling. Moreover, the metamodel's type system ensures that each interface is always referenced with its proper signature.

Illustration of DSL Elements: Modeling Examples

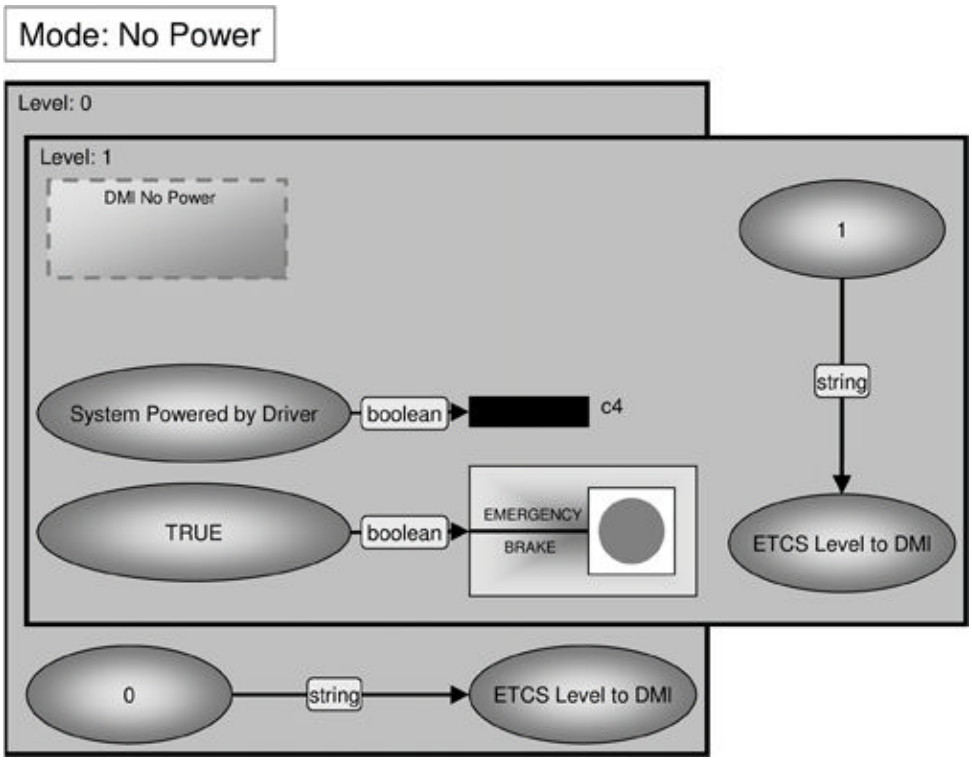
For illustration of the language elements we use a small case study presenting a highly reduced subset of the ETCS specification which is relevant for the EVC (UNISIG 2006a, Subset-26). It only uses EVC states which are needed to specify a minimal executable system in the ETCS application levels 0 (track is not equipped for ETCS) and 1 (track side-generated movement authority, continuous speed supervision, Eurobalises, for details see (UNISIG, 2006b, p. 19)). The example includes the states for full supervision and the handling of systems errors and operational threats, such as passing a stop-signal. A comprehensive description of the DSL sketched in this section, which also includes its formal metamodel, is given in (Feuser,& Peleska, 2011).

Table 1. EVC modes and transitions

	No Power (NP)	Stand By (SB)	Full Supervision (FS)	Staff Responsible (SR)	Unfitted (UN)	Trip (TR)	Post Trip (PT)	System Failure (SF)	Isolation (IS)
No Power (NP)		c4; p2							c1; p1
Stand By (SB)	c29; p2		c10; p7	c37, c8; p7	c60; p7			c13; p3	c1; p1
Full Supervision (FS)	c29;p5	c28; p5				c12, c16, c17, c20, c41, c65, c66; p4		c13; p3	c1; p1
Staff Responsible (SR)	c29; p2	c28; p5	c31, c32; p6		c21; p6	c18, c20, c36, c42, c43, c54, c65; p4		c13; p3	c1; p1
Unfitted (UN)	c29; p2	c28; p6	c25; p7	c44; p4		c39, c67; p5		c13; p3	c1; p1
Trip (TR)	c29; p2				c62; p3		c7; p4	c13; p3	c1; p1
Post Trip (PT)	c29; p2			c31; p4				c13; p3	c1; p1
System Failure (SF)	c29; p2								c1; p1
Isolation (IS)									

Table 1 shows the DSL representation of the reduced transitions table (the full table from (UNISIG, 2006e, p. 39) is a 16×16 matrix). The first column represents source modes, the first row target rows, and matrix element (m_1, m_2) specifies the guard condition to be fulfilled in order to perform mode transition $m_1 \rightarrow m_2$, as well as the priority applicable in situations where several guards evaluate to *true* in the current mode. Transition guards *g* are written syntactically in the form $g \equiv c_1, ..., c_n$ where the c_i are elementary conditions specified by the standard. The "," is interpreted as a disjunction, so guard *g* above evaluates to *true* if $c_1 \vee ... \vee c_n$ holds. Priorities are specified as *p*<*n*>where *n* is a natural number, and smaller numbers denote higher priorities. Comparison with the original transition table specified by the standard shows that the DSL variant displayed Table 1 is in on-one correspondence to its source (UNISIG, 2006e, p. 39). This facilitates the validation of the PIM against the standard. Observe, however, that the transitions table representation in a DSL modeling tools can be automatically transformed into suitable textual variants (for example, in XML format) that

Figure 6. Function block associated with mode No Power (NP)



can be interpreted by a code generator producing the associated state machine processing code in a high-level programming language.

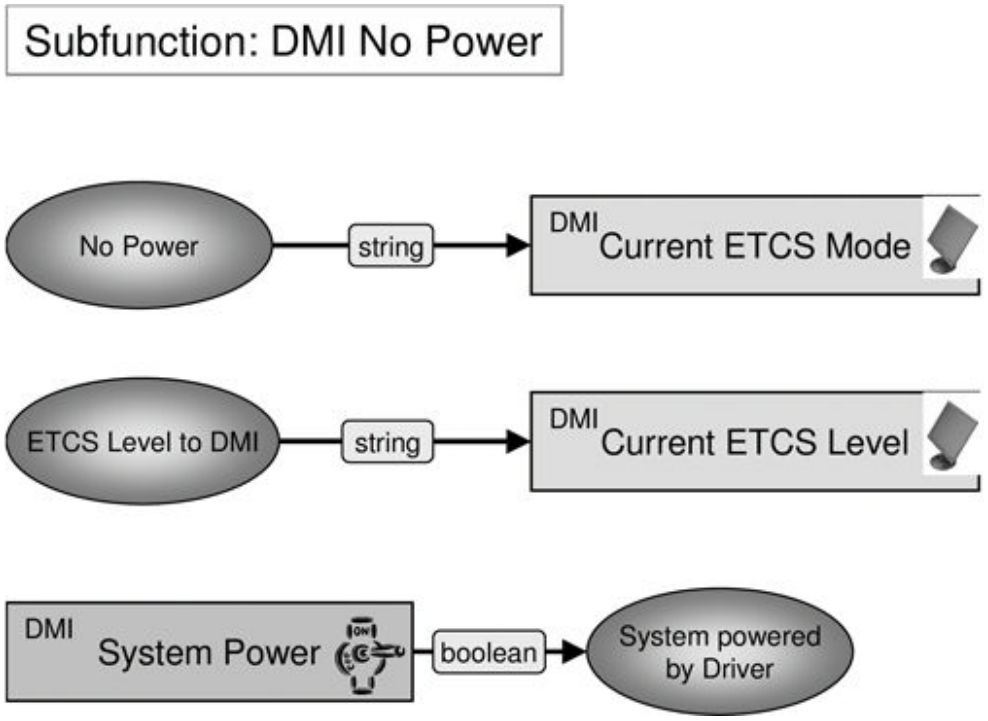
Function Blocks

The ETCS standard associates on-board EVC functions with modes. In our DSL this is reflected by the concept of *function blocks* linked with each mode. Consider the mode transition NP → SB which is guarded by condition c_4 according to the transitions table. The modes displayed in the first column/row are linked by the DSL to function blocks modeling the mode-dependent EVC behavior. For the NP mode this block is depicted in Figure 6(the DSL tool directly leads the designer from the transitions table to the selected mode's top-level function block). Function blocks can be decomposed in top-down fashion: in Figure 6, for example, the bold-lined rectangles represent the function block directly linked to mode No Power (NP), while rectangle DMI No Power refers to a subordinate function block whose details are depicted in Figure 7.

Signal Flow

Function blocks model the signal flow between constants (literals), model variables, guard (sub-) conditions and EVC interfaces (sources and sinks). Literals and variables are depicted as ellipses or circles, and EVC interfaces as rectangles decorated with interface-specific symbols. Examples of EVC interfaces captured in the DSL are the emergency brake (Figure 6), the driver machine interface in both communication directions (driver acts on switches and reads displays, Figure 7), and the odometer (Figure 9). The flow "*TRUE → EMERGENCY BRAKE*" in Figure 6, for example, models that the emergency brake

Figure 7. Function Block DMI No Power: Visualization of operational mode and level changes on DMI



is permanently engaged while the surrounding function block is active, that is, while mode NP applies. Flow arrows are decorated with type information in order to highlight the type conversions required by or resulting from operators combining and transforming several flows. Variable *ETCS Level to DMI*, for example, is the source for the display of the current ETCS application level at the driver machine interface. This variable is of type string since levels are defined as *0, 1, 2, 3, STM* (Specific Transmission Module, for communication with national ATP system, see (UNISIG, 2006b)). Therefore the flows transporting literal values 0, 1 to this variable are marked by *string*.

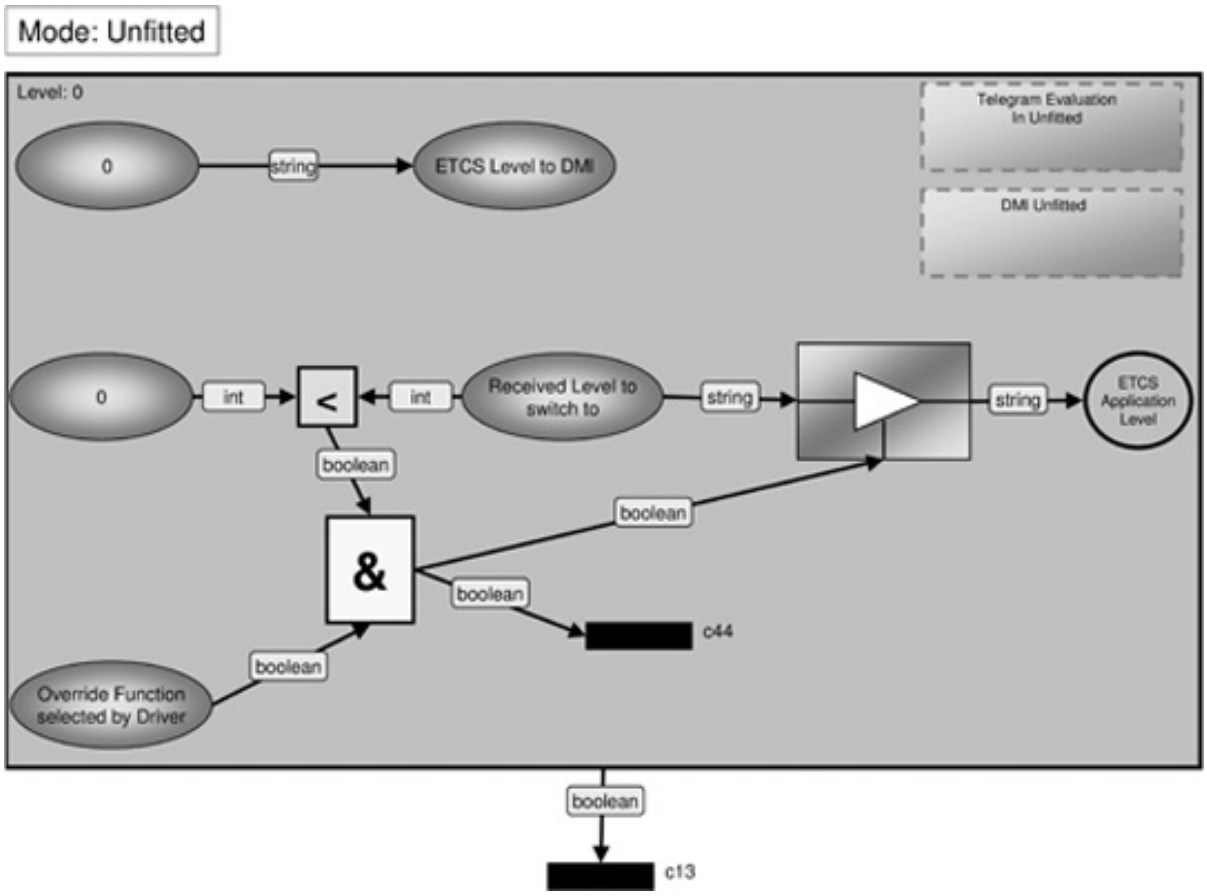
The conditions c_i referenced in the transitions table are managed in the DSL as special Boolean model variables which are represented as black horizontal bars, such as in the "*System Powered By Driver → c4*" flow in Figure 6: condition c_4 becomes *true* as soon as the EVC has been powered by the driver using the *driver machine interface (DMI)*.

Signal flows are executed permanently while the surrounding function block is active, that is, while the EVC resides in the mode associated with the top-level function block. In Figure 7, for example, the flow "*System Power → System Powered By Driver"* continuously copies the state of the power-on switch to model variable *System Powered By Driver*.

ETCS Application level dependencies

Mode-dependent function behavior may additionally depend on the ETCS level. This is reflected in the DSL by the possibility to use overlapping bold-line rectangles annotated with level numbers to model one function block, as depicted in Figure 6: in this example, the handling of the DMI (subfunction DMI No Power), as well the control of the emergency brake and of the condition c_4 are level-

Figure 8. Function block associated with operational mode Unfitted



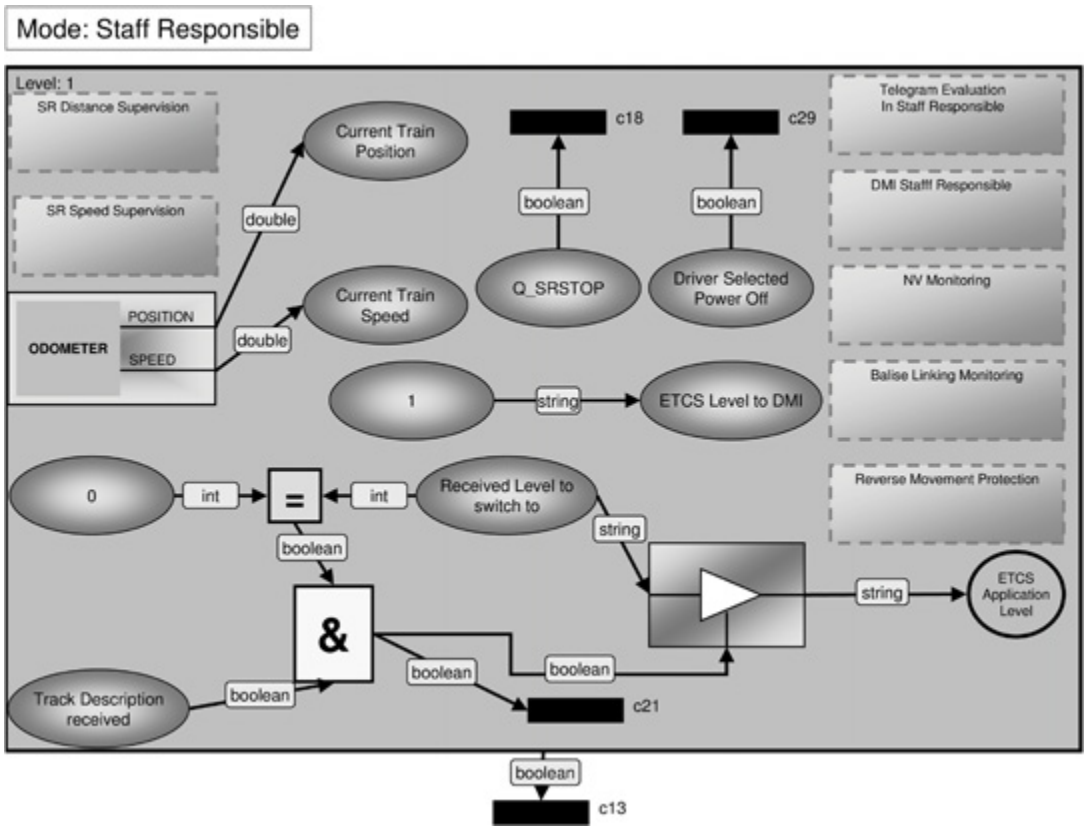
independent. Therefore the associated signal flows are located in the overlapping section of both rectangles. The setting of the *ETCS Level to DMI* variable, however, obviously depends on the level, so the different settings are depicted in the non-overlapping sections of each level-specific rectangle.

Simultaneous Mode and Level Transitions

Figure 8 and Figure 9 represent an example where a mode change is triggered by a simultaneous level change. Mode Unfitted applies to situations where the track side equipment does not support ECTS functionality, so this mode is only available in level 0. One of its main functions consists in the detection of a change to a higher level. This detection is based on the track side level identification which is communicated to the train via telegrams; the *Telegram Evaluation in Unfitted* sub-function places the level value received in the telegrams into model variable *Received Level to switch to.* A change in this variable, however, does not directly stimulate a level change in the EVC functionality: additionally, the driver has to select the override function via

The DMI. Formally, the signal flow specifies condition $b \equiv ((0 < \text{Received Level to switch to}) \wedge \text{Override Function selected by Driver})$. This condition specifies the truth value of guard condition c_{44}. Moreover, the inhibit gate marked by a $\triangleright$ symbol specifies that only when the b evaluates to *true* the new level value is published

Figure 9. Function block associated with operational mode Staff Responsible



as the new value of the *ETCS Application Level* variable. The transitions table specifies that in this case the mode changes to Staff Responsible (SR), whose associated function block is shown in Figure 9. Train operation in mode SR is performed under full responsibility of the driver who is "assisted" by the EVC by means of speed limit supervision, supervision of distances to potential danger points and detection of level changes (UNISIG, 2006d, pp. 19-21). The function block in Figure 9 shows how the mode may switch back to UN, if a track description is available and indicates in *Received Level to switch to* that the track side is again in level 0 condition (changes to levels > 1 are not shown since our example is restricted to levels 0 and 1). Variable *Q_SRSTOP* is updated by the telegram evaluation sub-function and indicates whether the train has to be stopped using the emergency brakes. This so-called *train trip* results in a transition to mode Trip (TR).

Exceptions

The ETCS standards require the EVC to detect safety-relevant equipment failures, and this shall lead to a transition into mode System Failure (SF) where the emergency brakes are engaged. Since the nature of equipment failures obviously is platform-specific, the requirement of the standard is generic, to be refined in the PSM created during a concrete development. Our DSL captures this situation in the PIM by providing the syntax for an *unspecified exception*, indicated by a Boolean flow emanating from the function block and setting guard condition c_{13} which forces the transition

to SF from any operational mode that was active before. The unspecified exceptions have to be fully specified on PSM level, by introducing concrete signal flows and associated control conditions for setting c_{13}.

Transformational Semantics and Model Validation

The DSL introduced above is associated with a metamodel formally specifying syntax and static semantics (Feuser, & Peleska, 2011). However, so far it is lacking a *behavioral* semantics specifying the set of computations (that is, sequences of valuations for interfaces and internal state variables) which can be performed by a given openETCS model. In principle, two approaches are available for associating behavioral semantics: (1) rules of an operational semantics are specified, showing how to generate a transition system from a given openETCS model or (2) a model-to-text transformation is specified that allows us to generate program source code with well-defined semantics from the model. In (2) the program semantics is simply lifted to *E* and *defined* to be the model semantics (see Fig. 11). Informally speaking, the "model *E* behaves as the program *P* generated from *E* does". The authors advocate the second alternative, because this facilitates the V&V work flow in a considerable way. Since the model semantics is by definition the one of the generated program, the model behavior can be directly explored by executing the program, preferably in a simulation environment on a host computer. Moreover, since the "program is the model", no equivalence or refinement proof is required to show that *P* is consistent with *E,* as would be required for variant (1) above.

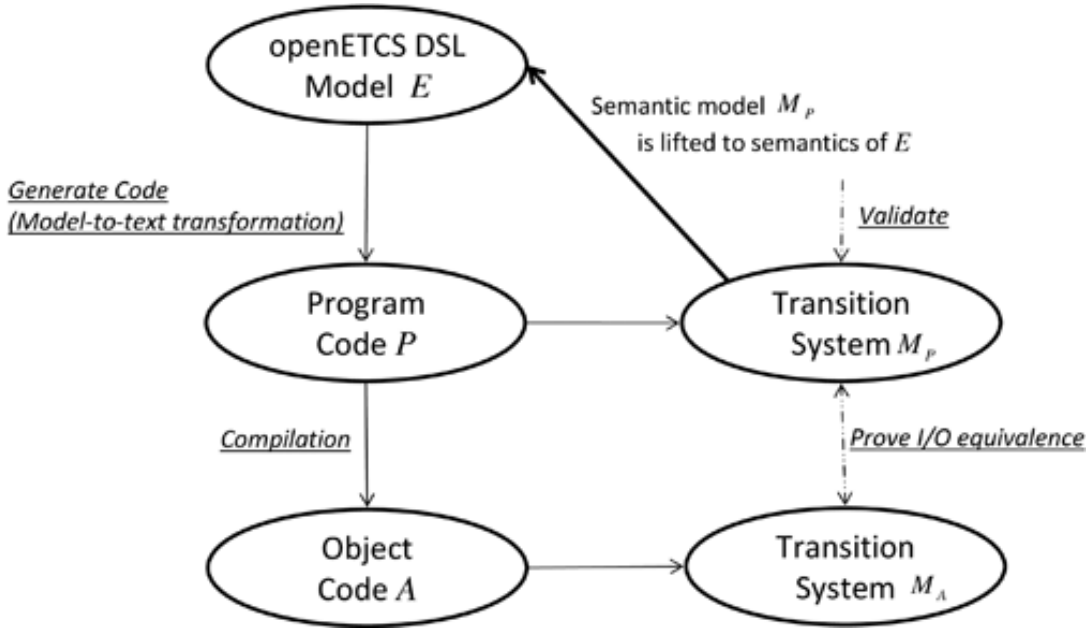
Observe that variant (2) is frequently used by model-based development tools: the simulators of the Matlab/Simulink and of the SCADE tools, for example, first create C code from models and generate programs from this code which can be executed for simulation purposes on host computers. The behavioral semantics of the original Matlab/Simulink model is that of the associated simulation program. As a consequence, design engineers usually explore the model by means of the simulation, in order to investigate whether the actual behavior associated with the model is really the one originally intended. A well-founded criticism to associating behavioral semantics by automated transformation into code is that the latter is often quite complex so that it would be practically infeasible to re-construct the operational semantics of a model from, say, the C program acting as its simulator. To address this objection, our approach is to use a semantically well-founded programming language: SystemC. For program code *P* written in SystemC a formal operational semantics has been defined. Moreover, SystemC not only provides the means for simulation, but also for model checking. This is described in more detail in Section *Object Code Verification.*

The advantages of a transformational semantics as described above in variant (2) obviously induce validation obligations for *P.* According to the requirements of the applicable standards it has to be shown that

1. *P* is free of run-time errors,
2. *P* is free of any malware imposing security threats,
3. *P's* behavior is consistent with the requirements of the applicable ETCS standards,
4. *P's* behavior is consistent with the particular development-specific requirements (this obligation only applies if *E* is the PSM of an EVC development).

Recall that *run-time errors* are violations of the programming language rules, potentially resulting in program behaviors that would be considered as failures for any set of applicable requirements. Typical examples of run-time errors are (1) arithmetic failures such as division by zero, over- and underflows, (2) de-referencing of null pointers, (3) reads

Figure 10. Transformational model semantics: Semantics of DSL model E is given by M_P



from and writes to illegal addresses, (4) violations of array boundaries (5) unintended endless loops, (6) violations of acceptable worst-case execution time (WCET) limits and, (7) stack overflows. Run-time errors could be caused by arithmetic modeling errors or erroneous array expressions on model-levels. Additionally, bugs in the code generator performing the model-to text transformation could introduce run-time errors without visible correspondence in the DSL model *E*. While in the case of a PIM the openETCS community is expected to contribute to the detection of run-time errors in a significant way, this would not suffice for receiving certification credit according to the standards (European Committee for Electrotechnical Standardization 2000, 2001a, 2003, 2001b, 2001c): according to today's state-of-the-art, it is strongly suggested to apply static program analysis by *abstract interpretation* in order to *prove* the absence of run-time errors in *P*. In the avionic domain this has become mandatory for maximum-criticality software; we expect that this will also apply to the railway domain in the near future, because powerful tool support is available (Cousot, Cousot, Feret, Mauborne, Miné, & Rival, 2009). Observe that absence of WCET violations and stack overflows cannot be verified on PIM level since they depend on the concrete underlying hardware architecture and require formalized hardware models in addition to the EVC models *E*.

The absence of malware would be ensured for PIMs by the openETCS community; for PSMs these threats have been analyzed in Section *Threat Analysis for Developments Following the openETCS Paradigm***.** It remains to consider methods to validate that the behavior of *P* is consistent with the informal requirements of the ETCS standard and that development-specific additions are properly reflected by the model.

For the validation of PIMs represented by programs *P* two techniques are available: model (property) checking and software testing. In both cases, *validation properties* or *test cases* including *expected results* are required, since a reference is needed for deciding whether the program behavior is correct. To this end the ETCS test plan and associated test cases serve as a starting point: as

described in (UNISIG, 2009) and the test case sub-documents referenced there, a collection of test cases applicable to ETCS functional requirements have been specified in structured natural language style. As far as applicable to the EVC, these test cases can be formalized to serve as properties to be verified against the model by means of model checking or as test cases with formalized stimulation sequences and assertions specifying expected results. On PIM level, these formalizations can be published again in order to let their consistency and completeness with respect to the ETCS be checked by public peer review. For a PSM the properties and assertions have to be extended by the manufacturer's V&V team in order to achieve full coverage of the PSM.

According to today's available technologies, verification against run-time errors of types (1,2,3,4,5) are performed on high-level code, that is, on the target of the model-to-text transformation. The same holds for most variants of model checking. In contrast to this, WCET analysis and stack analysis, as well as testing, check properties of the compiled program. For the purpose of model validation, analyses on high-level code are appropriate, because the model's behavior has to be explored, while its future embedding as binary image into some hardware platform is of no relevance. As a consequence, the high-level program *P* is adequate for model validation, and compilations into binary images are only performed in order to allow for dynamic testing of the model on host computers, in order to complement model checking. This leaves us with the problem to prove that the object code embedded into the target platform for concrete EVC developments is really consistent with programs *P* associated with platform specific models. This problem is addressed in the next section.

Object Code Verification

In this section, we outline a method for automated object code verification, applicable for (railway control) systems developed within a model-driven development framework as the one described above. The method is described in more detail in (Peleska & Haxthausen, 2007) and (Haxthausen, Peleska & Kinder, 2011).

Motivation For Object Code Verification

Automated object code verification for railway control systems is motivated by the fact that applicable standards for these safety-critical applications, e.g. the standard (European Committee for Electrotechnical Standardization, 2001a), require a substantial justification with respect to the consistency between high-level software code (e.g. C/C++ programs) and the object code generated by the associated compilers.

The conventional approach for such a justification is *compiler validation*: it is justified "once-and-for-all", and for any input (source code), that the compiler produces output (object code) which is a correct implementation of that input. However, these validations are far from being formal proofs, so errors may still be present in validated compilers. Strategies for fully formal compiler verification have actually been devised by several authors – see (Goos & Zimmermann, 1999) and the references listed there. However, to our best knowledge, formally verified compilers for the development of railway control systems are currently not used in practice. The reason for this is that the formal verification of industrial sized compilers is too time-consuming, while compiler updates occur quite frequently, so that a considerable number of non-trivial re-verifications would have to be performed.

An alternative or supplement to compiler validation is *object code verification*: each time object code is generated (by an arbitrary compiler), the generated object code is verified to be consistent with the source code. Object code verification has the advantage that it is independent of changes in the compiler, and hence re-verifications are not needed. Furthermore, it can be fully automated,

if the compiled code originates from high-level programs strictly adhering to certain programming patterns. While many arbitrary high-level programs do not comply to these patterns, they can be easily enforced for high-level programs automatically generated from abstract specification models. As a consequence, object code verification is considered by the authors as the preferred method to establish consistency between source and object code in the model-driven openETCS paradigm.

Object Code Verification Approach

In (Peleska & Haxthausen, 2007) and (Haxthausen, Peleska & Kinder, 2011) we suggested a specific approach to automated, formal object code verification, and we elaborated it for a railway case study where the source code consisted of SystemC programs automatically generated from abstract specification models, and the object code was GNU assembler code. SystemC was chosen because of its clearly defined semantics and the availability of model checking tools (Grötker, Liao, Martin & Swan, 2002). Our specific approach is as follows:

- An equivalence relation (*I/O equivalence* defined below) between source programs and object code is defined that characterizes consistent behavior at the interface level of the computer where the software is embedded.
- To prove that an assembler program (object code) A is a correct implementation of a source program P from which A had been compiled, A and P are mapped into associated behavioral models M_A and M_P, given in terms of some common semantic foundations (*I/O-Safe Transition Systems* to be explained below),
- Then it is proven that M_A and M_P are I/O equivalent by applying pre-defined transformations on M_A and M_P that have been proven "once-and-for-all" to preserve I/O behavior.

I/O-safe transition systems (IOTS) are similar to usual transition diagrams as defined e.g. in (Manna & Pnueli, 1992), consisting of nodes called *locations* and directed edges connecting them. One of the locations is designated (by only having an incoming arrow) as the initial location. The edges are labeled with guarded assignments (called *transition rules*) of the form [c]/x1:=e1,…,xn:=en, where the guard c is a Boolean expression and e1,…,en are right-hand side expressions of assignments to variables x1,…,xn. The variables associated with the IOTS are classified as being *input*, *output*, or *processing variables*, depending on whether they carry inputs from the environment to the system, outputs from system to environment or intermediate computation results, respectively. As a new feature of I/O-safe transition systems (compared to usual transition diagrams), locations are partitioned into pairwise disjoint sets of *input*, *output*, and *processing locations*, and there are further constraints on the allowed use of variables in guards and assignments in an IOTS: guards must only use processing variables; for edges into input locations the assignments must read input variables only and write to processing variables; for edges into processing locations assignments must read processing variables only and write to processing variables; for edges into output locations the assignments must read processing variables only and write to output variables. As a result of these restrictions, IOTS behavior can be structured into cycles consisting of input, processing and output phases. IOTSs are not affected by changes on input interfaces while processing a set of input vectors read before during the input phase. Analogously, the changes performed by IOTS components on output interfaces only become visible during the output phase, where all processing of the current input vector has already been completed. As an example, consider the following piece of SystemC code

```
for (int i=0; i<3; i++) A[i] = B[i];
```

where each A[i] is an output variable and each B[i] is a processing variable. This loop can be represented by the IOTS shown in Figure 11. In this diagram l3 is an output location and l0, l1, l2, and l4 are processing locations.

Compilation (using the gcc 4.0.2 compiler) of the for-loop shown above gives rise to the following object code (assembler) fragment:

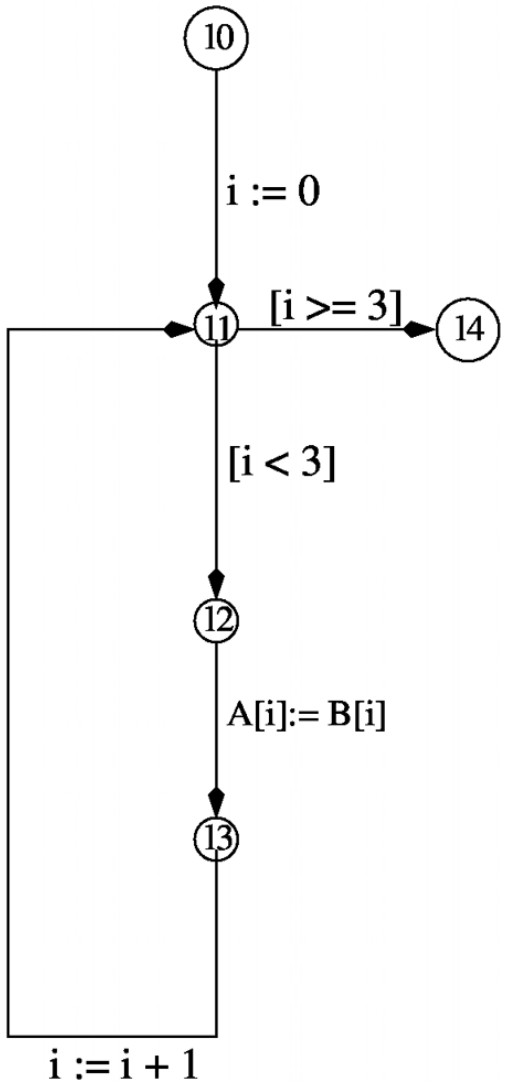
```
  movl $0, i
  jmp  .L103
.L104:
  movl i, %edx
  movl i, %eax
  movl B(,%eax,4), %eax
  movl %eax, A(,%edx,4)
  movl i, %eax
  incl %eax
  movl %eax, i
.L103:
  movl i, %eax
  cmpl $2, %eax
  jle  .L104
```

This can be represented by the IOTS shown in Figure 11. The new variable symbols eax, edx refer to registers, and ZF (zero flag), SF (sign flag) are flags indicating the outcome of the cmpl assembler instruction.

Intuitively, the object code is consistent with the source code, if their behavioral IOTS semantics are equivalent with respect to the inputs and outputs that can be observed in any execution of them. (In our context an output means an assignment to an output variable and an input means the assignment of the value of an input variable to a processing variable.) We will now define what it means to execute an IOTS and then define an appropriate equivalence relation between IOTSs.

A *state s* for an IOTS is an interpretation that assigns values to the variables of the IOTS. The *initial state* of an IOTS assigns initial values to each variable and assumes the initial location. A *computation* of an IOTS is a sequence of locations

Figure 11. Example of an IO-safe transition system diagram



l with associated states *s* satisfying the following requirements: (1) the first location and state are the initial location and state, respectively. (2) For each pair of consecutive pairs of locations and states *(l, s)* and *(l', s')* in the sequence, there is an edge in the IOTS from *l* to *l'* such that the guard of the edge evaluates to *true* in state *s,* and the execution of the assignments of this edge lead to state *s'*. The target state *s'* is the same as *s* except that (a) the interpretation of input variables is allowed to be

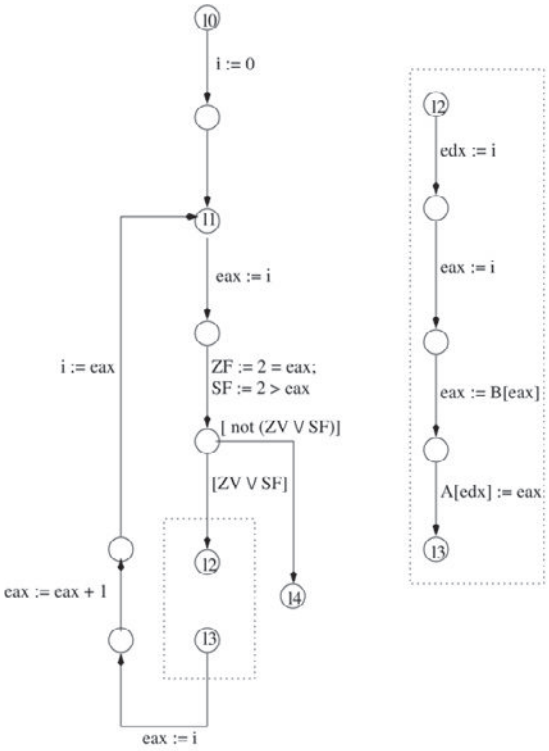
different (to model that the unrestricted environment can make inputs any time), and (b) the left-hand side variables occurring in the assignments of the edge carry their new values that are equal to the right-hand side expressions evaluated in *s*.

The *I/O observable behavior* of an IOTS computation is a sequence of states that is obtained by first restricting the given computation to a sequence of those states that belong to input and output locations, and then restricting these states to input and output variables only. Two IOTSs are *I/O equivalent* if a binary relation between their computations can be established, such that related computations have the same I/O observable behaviors modulo input/output mode-preserving variable renamings. The I/O equivalence of the two IOTSs shown in Figure 11 and Figure 12 can be proven by applying a sequence of graph transformations starting with the graph of Figure 11 and ending with the one of Figure 12. Each of the transformations applied preserves I/O equivalence. This transformation property can be proven once-and-for-all, so it does not depend on the specific compilation. As a consequence, I/O equivalence between source program *P* and object code has been established as soon as the sequence of graph transformations has been constructed. This construction process can be automated, and its underlying methods and algorithms are orthogonal to those used by the compiler. Therefore the probability that the compiler produces an object code error which is masked by an error in the object code verifier can be neglected.

Containing Security Issues Implied by Supplier-Specific Closed-Source Code

Consider a transformation where a given PIM is modified and extended to produce a PSM, as sketched in Figure 5. As indicated there, potential flaws in code resulting from modifications and extensions in the PSM may also compromise code

Figure 12. Example of an IO- safe transition system diagram for assembler code



originating from the PIM, by the way of covert channels or by sending corrupted data over the intended interfaces.

Partitioning

Protecting one component *C* against the corrupted behavior of other components *C'* is called *partitioning*. Techniques for enforcing partitioning are applied in several "dimensions": (1) in the data dimension, partitioning protects the internal state space of *C* from illegal writes originating from *C'*. (2) In the resource dimension, partitioning ensures that the specified amount of CPU bandwidth and communication interface bandwidth, as well as the pre-planned amount of statically allocated memory, heap and stack is assigned to *C*, even if *C'* tries to utilize an undue amount of these resources. (3) In the control dimension,

partitioning ensures that exceptions caused by corrupt behavior of *C'* will not affect *C*. (4) In the interface dimension, partitioning ensures that corrupt data passed from *C'* to *C* along intended interfaces does not compromise *C's* behavior.

Operating systems ensure property (1) by virtual addressing concepts supported by memory management units to be provided by the hardware. For (2), scheduling mechanisms allocating guaranteed CPU or channel bandwidth to processes are well understood, and have been implemented for example in the avionic domain as specified in (Aeronautical Radio Inc., 2005; Aeronautical Radio Inc., 2009). Property (3) is also ensured in operating systems like (Aeronautical Radio Inc., 2005) where processes (called *partitions* in the reference above) have their own data, heap and stack assignments, and only the causing process is affected by the resulting exception, and the scheduler ensures that only the admissible amount of CPU time is allocated to endless loops alternating between the causing and handling of exceptions. Property (4) is usually handled on application level where components have to perform integrity checks (from checksum calculations to logical tests) on incoming data.

These partitioning mechanisms are well-suited to protect re-usable code originating from the PIM when running together with platform-specific extensions. Observe, however, that partitioning requirements enforce the rule that any PIM component that has been modified or extended on PSM level shall be executed in its own address space which is separated from the one where unaltered PIM code runs in.

Hardware Virtualization

Providing partitioning for EVC software components under the regime of a single operating system has a severe drawback. Code generators for the PIM have to introduce some application interface (API) calls to an operating system, for example to protect critical sections by means of semaphores or to use communication interfaces. These API calls have to be mapped in the PSM to concrete calls to the operating system, without introducing new threats to the re-usable PIM code. We propose to use *hardware virtualization* for this purpose. In this scenario, EVC hardware platforms are equipped with a *hypervisor*, and different *guest operating systems* may run in *virtual machines (VMs)* accessing hardware and other resources through the hypervisor. We favor the concept of *para virtualization,* where the hypervisor has the capabilities of a micro kernel, so that no separate host operating system is necessary. Actions of guest operating systems requiring kernel privileges are mapped by the virtual machines to API calls of the hypervisor (see (Tanenbaum, 2008, pp. 568) for a more detailed overview). Such a hypervisor would assign strictly separated address spaces to different virtual machines and provide partitioning in the time domain by means of round-robin scheduling with fixed time slices for virtual machines, similar to the scheduler specified in (Aeronautical Radio Inc., 2005). The hypervisor would also provide driver management for hardware interface access with explicit assignment of interface visibility to selected virtual machines. Finally, the hypervisor would provide an inter-VM communication mechanism.

Software code generated from the PIM without alterations runs in a virtual machine of its own, using the API chosen for the platform-independent EVC models. Again, the API from (Aeronautical Radio Inc., 2005) would be a suitable candidate for this task. PSM-related code runs in a separate virtual machine. This concept can even be extended to several vendors providing different functional (for example, country-specific) adaptations: each of these adaptations would run in a separate virtual machine, thereby minimizing the possibilities for unwanted interference and providing the opportunity to suppliers to use their preferred operating systems.

Grey Channel Communication Paradigm

Since different virtual machines "see" each other as different computers, inter-VM communication can be regarded as a communication problem in distributed systems. Erroneous behavior of one virtual machine will only become visible to others on the designated inter-VM communication channels provided by the hypervisor. This situation, however, is well understood in the railway domain, and in (European Committee for Electrotechnical Standardization, 2001c) its implications are discussed and certification-related requirements are specified. Inter-VM communication is considered as an *open transmission system* communication, where the so-called *grey channel* may show any type of corrupted behavior (the possibilities are classified in (European Committee for Electrotechnical Standardization, 2001c)), and the communication end points are responsible for checking the integrity of incoming data.

Stack and WCET Analyses

Recall from the discussion Section *Transformation Semantics and Model Validation* that stack and WCET analyses have to be performed on machine code level. Observe, however, that stack analyses may be performed separately on process level, while WCET analyses may have to be performed on complete virtual machines and the applications deployed therein. Since by definition, processes P have their own stack and virtual address space, neighboring processes can never corrupt P's stack, provided that P is free of run-time errors resulting from memory copy instructions to the stack with illegal source lengths. This assertion has already been checked on source code level, and it has been shown by means of object code verification that the compiler does not introduce new run-time errors. In contrast to that, the scope of WCET analysis depends on the underlying scheduling policies: if processes $P_1, \ldots, P_n$ running in the same virtual machine are scheduled with guaranteed periods and CPU bandwidth, they may be analyzed separately. Otherwise they have to be analyzed jointly with their scheduler.

Virtualization and PIM Components

When structuring the PIM into a collection of sub-models, each classified as *never modifiable for PSM* and *modifiable for PSM,* another advantage of virtualization becomes apparent: each PIM sub-model can be transformed once-and-for-all into a binary image running with the API of the EVC operating system. In a concrete development, these binary images may be directly loaded into VMs offering this API.

Efficient Certification of openETCS Developments

A main advantage of the model-based openETCS approach with hardware virtualization consists in the possibility to re-use certification credit for unmodifiable PIM components, because they are embedded into VMs of the target platform as binary images without alterations. As a consequence, all tests and analyses (stack and WCET) performed before with the same image retain their validity because the VM guarantees the execution in the semantics of the EVC API. Observe that this advantage could not be exploited without hardware virtualization, because then a new compilation, link and load procedure would have to be performed for the generation of a new image suitable for the target operating system of the concrete development. Observe further, that re-use of the certification credit also depends on the capability of the hypervisor to enforce strict partitioning between different VMs with respect to all dimensions listed in the previous section.

Platform-specific components can be integrated into separate VMs, but require the full V&V effort applicable to new developments.

The drawback of having to provide certified hypervisors and virtual machines for a given EVC hardware platform is mitigated as soon as the hardware diversity is reduced and several suppliers contribute to one EVC development: this reduces the number of hypervisors to be developed and offers different suppliers to integrate EVC code without changes into new hardware platforms, because the platform offers a suitable VM.

FUTURE RESEARCH DIRECTIONS

It is our expectation that during the next years several competing DSLs for ECTS applications will be presented both by research and industrial communities. Their suitability will be assessed mainly from three perspectives: (a) the intuitive comprehensibility of ETCS requirements presented in the DSL under consideration, (b) the effort to develop code generators for the DSL and (c) the conceptual and tool support for PIM → PSM transformations.

Independent on the ETCS domain we also expect that object code verification will get increasing attention because wider use of model-driven development will enforce strict design patterns in generated code, so that the full capabilities of compilers will no longer be exploited by code-generating model-to-text transformations. It is even possible that code generator will directly transform from models to object code, because – since the software developers' focus is shifted from high-level code construction to DSL modeling – the advantages of expressive programming languages are no longer needed. In this case, however, the importance of model checking tools operating on object code level will considerably increase.

The model-based openETCS development paradigm and some of its major advantages depend on the availability of hypervisors and virtual machines allocating generated code on specific target hardware. The verification of hypervisors for certification purposes is a considerable challenge, but there already exist noteworthy successes in the field of automated verification (Leinenbach, & Santen, 2009). Due to the substantial interest in hardware virtualization in many domains we expect that both development and verification, and therefore also certification will be facilitated in the near future by following well-explored design and V&V patterns.

CONCLUSION

In this chapter a novel development and V&V paradigm was presented for the ETCS domain, with emphasis of development projects for the European Vital Computer EVC. The key characteristics of this paradigm are (1) a domain-specific modeling formalism applicable from EVC-related portions of the ETCS standard down to platform specific models to be transformed directly into binary images, (2) a strategy to apply hardware virtualization in order to facilitate the safe and secure integration of platform independent and platform specific code on a given hardware platform, (3) a comprehensive V&V approach leading to a higher re-use of certification credits for re-used model components, and (4) the quality benefits gained from a publication of models, code, V&V artifacts and an open tool chain facilitating the public peer review of EVC developments, at least for their platform independent parts.

Observe, however, that the publication of EVC development artifacts facilitates the certification process only in an indirect way: due to the fact that the applicable standards require formal documentation of deviations and concepts for their removal, the findings and subsequent corrections elaborated by the public reviewing process are only useful if documented according to the standards' requirements. Moreover, because certification credit is only given if reporters and maintainers can provide full evidence that their education and skills comply with the standards' requirements, all

public findings will have to be re-reviewed and documented by a professional core team.

ADDITIONAL READING

The general methodology of certification processes and the elaboration of safety cases has been described by Storey (1996). He also provides examples from different domains (with emphasis on avionic systems) that give a more intuitive understanding of these tasks than the standard (European Committee for Electrotechnical Standardization, 2000) which is applicable for ETCS-related developments. As indicated in Section *Transformational Semantics and Model Validation* above it is desirable in the model-driven openETCS paradigm that the high-level programming code generated from models should have a well-defined semantics and the availability of model checking tools. Apart from SystemC favored by the authors this is also possible for conventional programming languages, such as Java (Visser, Havelund, Brat, Park, & Lerda, 2003) or C (Beyer, Henzinger, Jhala, & Majumdar, 2007), if only restricted language subsets are used. Since automated code generators are applied in model-driven openETCS, these restrictions can be reliably enforced. In Section *Transformational Semantics and Model Validation* we stated that model checking of programs created by the code generator would typically be performed on high-level code. Schlich (2010) describes a model checking tool operating on machine code level. Its applicability, however, is currently restricted to microcontrollers of a smaller scale (in particular, smaller word size) than those that are suitable for EVC developments. Therefore these machine code level techniques are not yet suitable for the ETCS application domain. While we described a method for object code verification in Section *Object Code Verification* that relies on equivalence proofs of transition systems representing observable program behavior it should be noted that in specialized cases of model-based code generation also testing can be performed in an *exhaustive* way, so that passing a test suite implies I/O equivalence between model and object code (Löding, &Peleska, 2010).

Though the diversity of ETCS hardware platforms is presently quite large it should be noted that a similar diversity could be considerably reduced in the avionic domain, by enforcing the concept of *integrated modular avionics*. A small number of hardware suppliers provide a hardware platform that is standardized with respect to plugs, interfaces, operating system (Aeronautical Radio Inc., 2005) and configuration capabilities. Other suppliers deliver software running in separate partitions with the required protection as described in Section *Containing Security Issues Implied by Supplier-Specific Closed-Source Code*. The hardware/software integration is performed by the aircraft manufacture. For further details about IMA platforms and associated verification strategies see Efkemann, & Peleska (2011).

REFERENCES

Aeronautical Radio Inc. (2005). *ARINC 653, P1-2, Avionics Application Software Interface, Part 1, Required Services.*, Annapolis, MD: Aeronautical Radion Inc.

Aeronautical Radio Inc. (2009). *ARINC 664, P7-1, Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network.* Annapolis, Maryland: Aeronautical Radion Inc.

Berry, G. (2003). *The Effectiveness of Synchronous Languages for the Development of Safety-Critical Systems.* Retrieved April 3, 2011 from http://www.esterel-technologies.com/DO-178B/files/The-Effectiveness-of-Synchronous-Languages-for-the-Development-of-Safety-Critical-Systems.pdf

Beyer, D., Henzinger, T. A., Jhala, R., & Majumdar, R. (2007). The Software Model Checker Blast. *International Journal on Software Tools for Technology Transfer*, *9*(5-6), 505–525. doi:10.1007/s10009-007-0044-z

Cousot, P., Cousot, R., Feret, J., Mauborne, L., Miné, A., & Rival, X. (2009). Why does Astrée scale up? *Formal Methods in System Design*, *35*(3), 229–264. doi:10.1007/s10703-009-0089-6

Dingel, J., Diskin, Z., & Zito, A. (2008). Understanding and improving UML package merge. *Software and Systems Modeling*, *7*(4), 443–467. doi:10.1007/s10270-007-0073-9

Efkemann, C., & Peleska, J. (2011). *Model-Based Testing for the Second Generation of Integrated Modular Avionics*. To Appear in Proceedings of the A-MOST 2011.

European Committee for Electrotechnical Standardization. (2000). *EN 50126 - Railway applications - Communnications, signalling and processing systems – Specification and demonstration of reliability, availability, maintainability and safety (RAMS)*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001a). *EN 50128 - Railway applications - Communnications, signalling and processing systems - Software for railway control and protection systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001b). *EN 50159-1 - Railway applications - Communnications, signalling and processing systems – Safety-related communication in closed transmission systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001c). *EN 50159-2 - Railway applications - Communnications, signalling and processing systems – Safety-related communication in open transmission systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2003). *EN 50129 - Railway applications - Communnications, signalling and processing systems – Safety-relevant electronic systems for signaling technology*. Brussels: CENELEC.

European Railway Agency. (2007). *ERTMS/ETCS Functional Requirements Speci□cation FRS*. (Version 5.0)

Feuser, J., & Peleska, J. (2010). Security in Open Model Software with Hardware Virtualisation – The Railway Control Systems perspective. *Electronic Communications of the EASST, 33: Foundations and Techniques for Open Source Software Certification.*

Feuser, J., & Peleska, J. (2011). Dependability in Open Model Software with Hardware Virtualisation – The Railway Control System Perspective. Submitted to *Science of Computer Programming.*

Goos, G., & Zimmermann, W. (1999). Verification of compilers. In Olderog, E.-R., & Steffen, B. (Eds.), *Correct System Design, Recent Insight and Advances* (pp. 201–230). Springer-Verlag.

Grötker, T., Liao, S., Martin, G., & Swan, S. (2002). *System Design with SystemC*. Kluwer Academic Publishers.

Hase, K. R. (2009a). openETCS - Ein Vorschlag zur Kostensenkung und Beschleunigung der ETCS-Migration. *Signal+Draht 10*(10).

Hase, K. R. (2009b). openETCS - Open Source Software für ETCS-Fahrzeugausrüstung. *Signal+Draht*(12)12.

Hase, K. R. (2011). "Open Proof" for Railway Safety Software - A Potential Way-Out of Vendor Lock-in Advancing to Standardization, Transparency, and Software Security. In Schnieder, E., & Tarnai, G. (Eds.), *FORMS/FORMAT 2010 Formal Methods for Automation and Safety in Railway and Automotive Systems* (pp. 4–34). Berlin, Heidelberg: Springer-Verlag.

Haxthausen, A. E., Peleska, J., & Kinder, S. (2011). A formal approach for the construction and verification of railway control systems. *Formal Aspects of Computing*, *23*(2), 191–219. doi:10.1007/s00165-009-0143-6

Kelly, S., & Tolvanenm, J.-P. (2008). *Domain-Specific Modeling*. Hoboken, New Jersey: John Wiley & Sons Inc. doi:10.1002/9780470249260

Kent, S. (2002). Model driven engineering. In *Proceedings of the Third International Conference on Integrated Formal Methods, IFM '02*. 286-298, London, UK: Springer-Verlag.

Leinenbach, D., & Santen, T. (2009). Verifying the Microsoft Hyper-V Hypervisor with VCC. In Cavalcanti, A.,& Dams, D. R. (eds.), *Proceedings of the 2nd World Congress on Formal Methods*, (pp. 806-809), Berlin Heidelberg: Springer-Verlag.

Löding, H., & Peleska, J. (2010). Timed Moore automata: test data generation and model checking. In *Proceedings of the Third International Conference on Software Testing, Verification and Validation ICST,* (pp. 449-458). DOI http://doi.ieeecomputersociety.org/10.1109/ICST.2010.60

Manna, Z., & Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag. doi:10.1007/978-1-4612-0931-7

Mewes, K. (2010). *Domain-specific Modelling of Railway Control Systems with Integrated Verification and Validation*. München: Verlag Dr. Hut.

Object Management Group. OMG (2010a). *OMG Unified Modeling Language (OMG UML), Infrastructure, V2.3.* Retrieved April 1, 2011, from http://www.omg.org/spec/UML/2.3/

Object Management Group. OMG (2010b). *OMG Unified Modeling Language (OMG UML), Superstructure, V2.3*. Retrieved April 1, 2011, from http://www.omg.org/spec/UML/2.3/

Peleska, J., & Haxthausen, A. E. (2007). Object code verification for safety-critical railway control systems. *Formal methods for automation and safety in the railway and automotive systems (FORMS/FORMAT 2007).* Braunschweig, Germany: GZVB e.V.

Platzer, A., & Quesel, J.-D. (2009). European Train Control System: A Case Study in Formal Verification. In K. Breitman & Cavalcanti, A. (eds.), *11th Conference on Formal Engineering Methods ICFEM 2009*: LNCS 5885 (pp. 246-265). Berlin, Heidelberg: Springer.

Pnueli, A., Shtrichman, O., & Siegel, M. (1998). The Code Validation Tool CVT: Automatic verification of a compilation process. *International Journal on Software Tools for Technology Transfer*, *2*(1), 192–201.

Schlich, B. (2010). Model Checking of Software for Microcontrollers. *ACM Transactions in Embedded Computing Systems*, *9*(4), 1–27. doi:10.1145/1721695.1721702

Schmidt, D. C. (2006). Model-Driven Engineering. *IEEE Computer*, *39*(2), 25–31. doi:10.1109/MC.2006.58

Stallings, W. (2008). *Operating systems: internals and design principles*. Upper Saddle River, NJ: Prentice Hall.

Storey, N. (1996). *Safety critical computer systems*. Upper Saddle River, NJ: Prentice Hall.

Tanenbaum, A. S. (2008). *Modern Operating Systems*. Upper Saddle River, NJ: Pearson.

Trowitzsch, J., & Zimmermann, A. (2006). Using UML state machines and petri nets for the quantitative investigation of ETCS. In *Proceedings of the 1st international conference on performance evaluation methodolgies and tools* (pp. 34–es).

UNISIG. (2006a). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-1*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006b). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-2*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006c). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-3*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006d). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, -026-4*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006e). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-5*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006f). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-7*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006g). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-8*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2009). */ETCS – Class 1 Test Plan. 076-0*(2.3.1). Retrieved April, 9, 2011, from http://www.era.europa.eu/Document-Register/Pages/UNISIG%20SUBSET-076-0.aspx

Visser, W., Havelund, K., Brat, G., Park, S., & Lerda, F. (2003). Model Checking Programs. *International Journal on Automated Software Engineering*, *10*(2), 203–232. doi:10.1023/A:1022920129859

Zimmermann, A., & Hommel, G. (2005). Towards modeling and evaluation of etcs real-time communication and operation. *Journal of Systems and Software*, *77*(1), 47–54. doi:10.1016/j.jss.2003.12.039

zu Hörste, M., & Schnieder, E. (1999). Modelling and simulation of train control systems using Petri nets. In FMrail workshop (Vol. 3).

KEY TERMS AND DEFINITIONS

API: Application Program Interface to an operating system

ATO: Automated Train Operation

ATP: Automated Train Protection

Compiler Validation: Activity for ascertaining that a compiler accepts the source code compliant with the specified syntax of the input language and produces outputs (e.g. object code or code represented in any other formal language) which is consistent with the semantics of the source code.

Covert Channels: A communication channel, which has not been intended in the original software design, but may be used by a corrupted software component to compromise the behavior of another component.

DMI: Driver Machine Interface

DSL: Domain Specific Language

ETCS: European Train Control System, a standardized approach to on-board control systems and track side systems allowing trains to travel across boards in Europe.

EVC: European Vital Computer, the on-board controller responsible for automated train protection and automated train operation

PIM: Platform Independent Model, a model specifying structural and behavioral properties of a system without identifying the concrete hardware platform and operating system environment where the application software specified by the model is to be deployed.

PSM: Platform Specific Model, a model where the operating system and associated API, drivers and hardware have been fully specified in addition to the information already provided by a PIM. The PSM contains all the information required for hardware/software integration of the system to be developed.

Wide Spectrum Formalism: Modeling formalism which is suitable for a wide range of application domains. This term is used as a contrast to DSLs that were explicitly designed to support only a well-defined application domain, but with a higher degree of intuitive syntax and pre-defined semantic relationships that are characteristic for the domain.

Section 2

Hazard Analysis and Model-Based Evaluation

Chapter 3
Semi-Quantitative Risk Assessment of Technical Systems on European Railways

Jens Braband
Siemens AG, Germany

ABSTRACT

The European Railway Agency (ERA) has the challenging task of establishing common safety targets and common safety methods throughout Europe. In this context, the harmonization of risk analysis methods is also discussed. The purpose of this paper is to present a new semi-quantitative approach for the risk analysis of technical systems and the means by which compliance with legal and regulatory requirements can be demonstrated. As a particular reference, a new German pre-standard, which lays out requirements for semi-quantitative approaches, is taken into account.

INTRODUCTION

The European Railway Agency (http://www.era.europa.eu), established by European Regulation 881/2004, has the mission of reinforcing railway safety and interoperability throughout Europe in times of ongoing privatization. Central to its work on railway safety is the development of measures based on common safety targets (CSTs) and common safety methods (CSMs), common safety indicators and harmonized safety certification documents.

The common safety methods describe how safety levels, the achievement of safety targets and compliance with other safety requirements are assessed in the various member states. As a first step, EC Regulation 352/2009 will finally come into force for the complete railway sector by July 2012. In this regulation, a semi-quantitative risk acceptance criterion for technical systems (RAC-

DOI: 10.4018/978-1-4666-1643-1.ch003

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

TS) similar to civil aviation has been introduced: *For technical systems where a functional failure has credible direct potential for a catastrophic consequence, the associated risk does not have to be reduced further if the rate of that failure is less than or equal to 10^{-9} per operating hour*.

This criterion is limited to those technical systems where failure can lead to catastrophic effects, e.g. accidents involving several fatalities, and for which there are no credible barriers or substantial mitigating factors that will prevent this consequence from materializing. The criterion can be used for the most critical functions performed by technical systems on railways such as speed supervision, control of the switch position, complete and permanent loss of the brake system, or loss of the traction cut-off function. This means that formally RAC-TS relates only to potentially catastrophic accidents. In order to apply it also to other severity categories, RAC-TS has to be embedded in a risk analysis method.

The chapter is organized as follows: after a description of problems with risk analyses, an applicable standard is reviewed, from which the requirements are taken. Then a new semi-quantitative risk analysis method is constructed and some arguments and examples concerning the validation of the method are presented. Finally directions for future research are given.

Problems with Risk Analyses in Railway Applications

Risk is a combination of accident severity and accident frequency. Accident frequency can be calculated by hazard frequency and the probability of a hazard developing into an accident. This probability is derived by taking into account the efficiency of barriers. Barriers can be of different origin, e.g. human interactions, operational barriers, technical barriers.

It is well known that risk acceptance is an intricate topic and that risk analyses may be quite time-consuming and tedious [Braband (2005)], in particular if they are performed quantitatively. There exist simpler semi-quantitative methods, e.g. risk matrix, risk graph or risk priority numbers, however they often lack justification and it is not clear whether the derived results are trustworthy. So, a major research challenge consists in constructing dependable semi-quantitative methods.

In particular, schemes based on risk priority numbers (RPN) are widely used in Failure Modes, Effects and Criticality Analyses (FMECA) although it is known that they have not been well constructed and that their use may lead to incorrect decisions:

- The risk of different scenarios that lead to the same RPN may differ by orders of magnitude
- Scenarios with similar risks lead to different RPN

This has already been observed by Bowles (2003) and has now also lead to cautionary advice in the standards.

Risk matrices are a well-known tool in risk assessment and risk classification, also in the railway domain (see for example EN 50126 (1999) or Braband (2005)). Table 1 gives a typical example. The major drawbacks of such risk matrices are:

- Risk matrices must be calibrated to their particular application.
- The results depend on the system level to which they are applied.
- The parameter classes must be concisely defined in order to avoid ambiguity and misjudgments.
- It must be defined which frequency is meant, e.g. accident or hazard frequency.
- It is not directly possible to take barriers or risk reduction factors into account in the risk matrix.

However, if these drawbacks can be overcome, risk matrices are a well-accepted and easy-to-

Table 1. Typical risk matrix

Frequency	Severity			
	Negligible	**Marginal**	**Critical**	**Catastrophic**
Often	10	6	3	1
Probable	14	9	5	2
Occasional	18	13	8	4
Rare	21	17	12	7
Improbable	23	20	16	11
Unbelievable	24	22	19	15

use tool, also for risk prioritization (see the rank numbers in Table 1).

If risk matrices are to be applied in the railway domain, they need to be applied in combination with a method which can additionally take into account the effect of barriers and their risk-reducing effect. Typical candidates would be the fault tree analysis (FTA) in a quantitative analysis or semi-quantitative tables as used by risk priority numbers.

In conclusion for the railway domain semi-quantitative methods are very attractive and already widely used, but their dependability is questionable. Only a few approaches (see Bepperling (2008) and Milius (2010)) have been presented so far where semi-quantitative methods have formally been validate. But a standard for the use of such methods or against which methods can be checked has been missing so far.

CONSTRUCTION OF A SEMI-QUANTITATIVE RISK ANALYSIS METHOD

Requirements for Semi-Quantitative Risk Analysis Methods

Although semi-quantitative risk analysis methods are very popular in many application areas, they have been justified only informally. Requirements for such methods were not clear in the past, but recently a German pre-standard DIN V VDE V 0831-101 has clearly set out the requirements. So, it is now possible to construct a method and validate it with respect to these requirements. There are in total 28 requirements, but not all relate to construction of the method. Table 2 gives an overview of the requirements; the mandatory requirements appear in bold. For more details we have to refer to DIN (2011).

Risk Score Matrix Approach

A semi-quantitative approach is proposed which fulfils all requirements of the German pre-standard DIN V VDE V 0831-101. The model is called Risk Score Matrix (RSM) and consists of the application of a risk matrix and semi-quantitative score tables for assessment of the barriers. The complete approach is shown in Figure 1, jointly with additional and alternative steps. The final result consists of the hazard rates (HR) related to the functional failures (as hazards) of the technical system and the assumptions on which the analysis rests, which may turn into safety-related application rules (SAR).

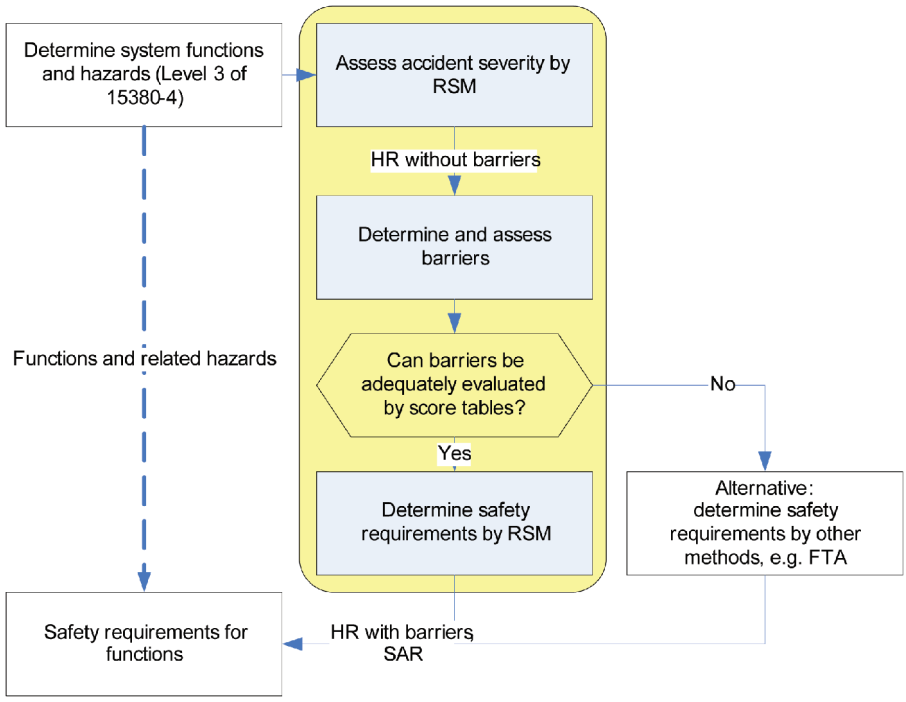
System Definition

The discussion in this paper focuses on technical systems only. According to EU Regulation 352/2009, a technical system is a product developed by a supplier including its design, imple-

Table 2. Summary of requirements

Construction	A1	State clearly reference units and application scope.
	A2	Be conservative in your assessment.
	A3	Make sure your parameter granularity is sufficient.
	A4	Work out a user guide.
	A6	State clearly the system level to which the method applies.
	A8	Allow for hazard classification.
	A12	Assessment of accident severity
	A13	Assessment of accident frequency
	A14	Description of all barriers
	A15	The tables should be compatible.
	A17	Assessment of human reliability
	A18	Assessment of operational barriers
	A19	Assessment of exposition
	A20	Assessment of external barriers
	A21	Assessment of technical barriers
	A22	Take into account dependencies of barriers.
	A23	Calibrate the method (against a risk acceptance criterion).
	A24/ A25	Assure proportionality between risk and criticality.
	A26	Small changes in the parameters lead to small changes in the result.
	A27	A safety requirement has to be derived.
	A28	Give rules on how to derive the Safety Integrity Level.
Application	A5	Justification of parameter choice
	A7	Identify hazards systematically.
	A9	Work out hazard scenarios.
	A10	Justify the choice of the relevant scenario.
	A11	Document results in a hazard log.
	A16	Identify safety-critical application conditions.

Figure 1. Overview of the Risk Score Matrix model



mentation, and support documentation. It should be noted that:

- The development of a technical system starts with its system requirements specification and ends with its safety approval.
- Human operators and their actions are not included in a technical system, however their actions may be taken into account as (external) barriers mitigating the risk
- Maintenance is not included in the definition, but in maintenance manuals.
- Technical systems can be subject to a generic type approval, for which a stand-alone risk acceptance criterion is useful.

A function is a "specific purpose or objective to be accomplished that can be specified or described without reference to the physical means of achieving it." A function level is defined in prEN 15380-4 (2010) as "level, to group functions of equal purpose". The distinction between levels is described informally as follows:

- First-level function: functional domain that encompasses a set of functions related to the same general focus or service for the considered (rolling stock) system.
- Second-level function: related to a specific set of activities which contributes to completion of the functional domain defined at the first level (at this level, it is not said how a second-level function is to be implemented).
- Third-level function: related to a specific activity out of the related set of activities, it encompasses a set of tasks (a function at least at level 3 should be supported as much as possible by one single subsystem).

It is proposed to use prEN 15380-4 (2010) which contains up to five hierarchical levels. Taking into account the definition of function level, level 3 seems to be the most appropriate for the application of RAC-TS. At least it does not seem reasonable to go into more detailed levels such as level 4 or 5. Table 3 gives some examples of functions to which RAC-TS may be applied. Although prEN 15380-4 (2010) relates to rolling stock only, it can be extended to infrastructure functions quite easily, e.g. by identification of all interfaces of other functions to rolling stock. Some functions (or at least interfaces) are already included in group K. In Table 3, some examples of level 3 functions related to signaling are proposed.

Table 3. Examples of signaling functions

Code	Function description
=LBB	Detect track vacancy
=LBC	Detect train at a particular spot
=LBD	Locate train
=LCB	Determine train description
=LDB	Provide diagnostics
=LEB	Supervise driver vigilance
=LEC	Automatic train stop
=LED	Supervise braking curve
=LEE	Supervise maximum train speed
=LFB	Optimize train running
=LGB	Monitor switch
=LGC	Lock switch
=LGD	Monitor derailer
=LGE	Lock derailer
=LGF	Monitor grade crossing
=LHB	Provide signal information
=LJB	Provide cab radio
=LKB	Display state to driver
=LKC	Display state to dispatcher
=LKD	Transmit commands

Risk Matrix

A suitable risk matrix has already been proposed and justified in Braband (2011), see Table 4. The table shows intolerable and tolerable combinations in a frequency scaling of $\sqrt{10}$. Safety targets

Table 4. Proposed risk matrix

HR	A	B	C	D	E
>10^{-5}/h					
10^{-5}/h			Intolerable		
$3x10^{-6}$/h					
10^{-6}/h					
$3x10^{-7}$/h					
10^{-8}/h					
$3x10^{-8}$/h					
10^{-8}/h		Tolerable			
$3x10^{-9}$/h					
10-9/h					RAC-TS

Table 5. Consolidated severity categories

ID	Combinations	FWI range	Typical FWI
E	Multiple fatalities	2≤FWI	5
D	Single fatality or multiple serious injuries	0.2≤FWI<2	1
C	Single serious injury or multiple light injuries	0.02≤FWI<0.2	0.1
B	Single light injury	0.01≤FWI<0.02	0.01
A	-	FWI<0.01	n. a.

would be chosen at the boundary between these two regions (medium gray shading). This scaling is compatible with the common scaling for Safety Integrity Levels (SIL), as two classes from one SIL.

The corresponding accident severities are defined in Table 5. Classification can be performed based on a qualitative estimate of the typical accident severity or based on statistical data (fatalities and weighted injury score (FWI)). Note that "typical" does not mean worst case; in a safety sense, it should be interpreted as a typical bad outcome, i.e. worse than average, e.g. 90% percentile.

Assessment of Barriers

The model takes into account the following types of barriers:

- Possibility to avoid accident by human interaction (H)
- Possibility to mitigate the hazard by an independent technical system (T)
- Operational environment (B)
- Demand frequency assessment (D)

The presence and efficiency of these barriers together with the severity assessment determine the outcome of the assessment and thus the appropriate safety requirements that will have to be achieved for the technical system under study. The assessment is carried out via a score scheme where scores are allocated to the barriers and then these scores are added to calculate the total risk reduction, starting from the risk matrix in Table 4. Since the scores for the barriers are added instead of multiplied, this means that the scores allocated are given in a logarithmic scale where each score

Table 6a. Action type assessment

A – score	Action type	Comment
4	Skill-based	Well-known and trained skill-based action
2	Rule-based	Rule-based action that has been appropriately trained and managed
0	Knowledge-based	But no routines or rules are defined.

Table 6b. Work environment assessment

W – score	Work environment	Comment
1	Good conditions	The work is performed under normal conditions with regard to sight, noise, physical forces and weather.
0	Adverse conditions	The working conditions are adverse with regard to at least one factor: lighting, noise, physical forces (e.g. excessive vibrations) or adverse weather conditions (too cold, too hot, etc.).

Table 6c. Stress level assessment

ST – score	Stress level	Comment
1	Optimal	
0	Excessive demands	The work load is very demanding. The stress level is high, e.g. work under time pressure.
	Insufficient demands	The work performed is not very demanding and mostly routine.

represents a "risk reduction" with a factor of $\sqrt{10}$ and two scores represent a reduction of one order of magnitude (i.e. one SIL).

The complete risk reduction is then calculated as the sum of scores, possibly reduced by a score accounting for the level of independence of the different barriers present. This is to avoid adding several barriers that are functionally dependent on each other and that are likely to fail simultaneously.

It should be noted that such a semi-quantitative assessment method may not fit all particular problems; e.g. there may be rare cases when other barriers occur and need to be taken into account. Also, some of the tables may be overly conservative, e.g. the assessment of human reliability by parameter H. In such cases, it is advised to apply first the risk matrix (Table 4) without any barriers and evaluate the barriers by an alternative method, e.g. Fault Tree Analysis, Event Tree Analysis or Markov models, as appropriate for the particular problem.

For the sake of brevity, it is not possible to present and discuss all score tables. Instead, the focus will be on two often used table types.

Assessment of Human Reliability

In some situations, it can be foreseen that there are still barriers present after the failure of a technical system due, for example, to the driver or staff observing the problem and acting correctly. Human interaction can also, in some cases, be carried out by passengers or third persons. Examples could be staff or passengers correctly using on-board fire extinguishers in case of fire or similar situations. Evaluation is based on three Tables 6a, 6b and 6c and calculates a combined score as the sum of the following sub-scores:

Table 7. Examples of operational environment mitigation barrier scores setting

B – scores	Traffic density	Description
3	Much less dense than average	Scarcely used lines
2	Less than average	Less used networks such as regional lines
1	Average	Network-average conditions
0	Above average	Condition above network average (e.g. high-speed lines)

- Type of task
- Stress level at which the task is performed
- Environmental conditions under which the task is performed

The approach is similar to simple screening techniques in human reliability assessment, e.g. Accident Sequence Evaluation Program (ASEP), e.g. Sträter (1997) or the approach validated by Hinzen (1993). Such approaches are known to be pragmatic and generally conservative. Note that also alternative assessment schemes could be transformed into similar tables. This assessment of human barriers does not pretend to give a deep and exact description of the human actions to be carried out and their reliabilities. It merely intends to give a conservative order estimate and does not replace further ergonomic studies, e.g. on the design of human-machine interfaces.

Pre-conditions for the application of this assessment are:

- Operators must be properly trained and have sufficient experience.
- There must not be any goal conflicts in performing the task, e.g. safety vs. performance.

The combined score is then calculated from Tables 6a, 6b and 6c as H= A + W + ST.

Assessment of Operational Barriers

The operational environment may influence the probability that the failure of a technical system develops into an actual accident with consequences for human safety. Basically, these factors depend on the "level of use" or "operational density" and may vary depending on the type of system that is under study. Typically, they could relate to passenger density on trains, road / rail level of traffic at a grade crossing to determine collision risks, railway traffic in the network to determine the probability that a track section is occupied, etc.

Table 7 gives an example for barriers whose efficiency depends on traffic density. Obviously, this is not the case for all types of operational environment barriers, which is why also the applicability of this table needs to be assessed before using it.

Assessment of Barrier Dependency

For every barrier that is taken into account, it must be analyzed whether its risk reduction is independent from the previous barriers. If it is not so, some scores will be subtracted from the score of the barrier, in accordance with Table 8b below. If the correlation is strong, the new barrier may reduce the risk only marginally.

Tables 8a and 8b can be justified on the basis of experience with conditional failure probabilities in human task analysis, e.g. Sträter (1997) and common cause analysis of technical systems.

The reduction of the barrier score is calculated by Table 8b, which gives the reduction of the barrier score Φ as a function of the original barrier score (top row) against the dependency of the new barrier with respect to all previous barriers.

Table 8a. Dependency classes

Dependency class	Comment
Independence (I)	There is no functional dependence between the factors; no common causes for failures exist.
Low dependence (LD)	The barriers are statistically independent; no significant physical influence. Related to human tasks, the task is performed by a different person at a different location and in a different operational situation.
Medium dependence (MD)	The mitigating factors have a single common cause failure – if one barrier fails, there is a slightly increased chance that the other also fails. Related to human tasks, e.g. two of the following characteristics are the same: same person, same location or same operational situation.
High dependence (HD)	The barriers have more than one common cause. If one barrier fails, there is a significantly increased chance that the other also fails.
Complete dependence (CD)	Several common causes. The new barrier will not be taken into account.

Table 8b. Dependency assessment

Φ	1	2	3	4	4+i
I	0	0	0	0	0
LD	0	0	-1	-1	-(i+1)
MD	0	-1	-1	-2	-(i+2)
HD	0	-1	-2	-3	-(i+3)
CD	-1	-2	-3	-4	-(i+4)

Validation of the Risk Score Matrix Method

It is not possible to give all arguments concerning the requirements from Table 2 here, but it is possible to sketch a few of the key arguments, whose fulfillment is quite obvious by the construction of the tables. For examples of the complete validation of semi-quantitative approaches, see Bepperling (2008) and Milius (2010).

The scope as well as the units of measurement are well defined by Table 3 and RAC-TS, so A1 and A6 can be fulfilled. As all tables are constructed conservatively, A2 is met. The granularity of the method is set to √10, which fits well to the SIL scale and is reasonable, so A3 can be fulfilled. As this scaling is used consistently throughout all tables, A15 is complied with. The tables shown in this section also meet the respective requirements A12, A13, A17, A18 and A22. The method is also calibrated appropriately against RAC-TS, so A23 follows. The method is monotonous with respect to risk (A24), i.e. a higher risk gains a more demanding safety requirement. Also, small changes in the parameters lead only to small changes in the safety requirements (A26).

Examples

In some cases, like =LGB from Table 3, RAC-TS is directly applicable. The main hazard would be that the status of a switch would be determined wrongly so that a train may run over a switch which is set in an incorrect direction. If passenger trains at high speed ran over this switch, then ID E would be determined from Table 5 leading to a THR of 10^{-9} per operating hour per switch. Some human mitigation may be possible (e.g. at low speed) and there is also the possibility that the switch is not set in the branching direction (50% chance), so that the overall score (due to Tables 6a to 6c) may be assessed as 1, leading to a THR of $3x10^{-9}$ per operating hour per switch.

Table 9. Examples of accident classification

ID	Derailment...	Collision...	Impact	Personal accidents
E	Passenger train at high speed	Passenger train on a main line		-
D	Passenger train at medium speed	At a grade crossing	Train with track gang	Passenger falling out of a train at high speed
C	Passenger train at low speed			Passenger falling out of a train at low speed or when train has stopped
B	In shunting operation		Train into buffer at low speed	Passenger hit by a door Passenger falling during embarkment
A				

In another example, =LGF from Table 3, the main hazard would be that road traffic would not be protected by the grade crossing and the consequence might be a collision at the grade crossing, from which ID D as the typical accident severity would be derived from Table 5 leading to a THR of 10^{-8} per operating hour per grade crossing. Additionally, human mitigation may be possible (e.g. at low speed or with good sight) by the road users, so that the score (due to Tables 6a to 6c) may be assessed as 1. However, this mitigation is not independent from the severity estimate. Additionally, it can be taken into account that grade crossings are not allowed on high-speed lines and often avoided on lines with high traffic density. Thus, finally a score of 1 may be assessed, leading ultimately to a THR of $3x10^{-8}$ per operating hour per grade crossing.

FUTURE RESEARCH DIRECTIONS

The list of functions to which RSM shall be applied has only been identified by general principles or by examples, e.g. Table 3. Such a function list has to be elaborated in full detail. The assessment of accident severities in Table 5 can be further improved in order make the assessment more user-friendly. An example of a simple classification of accidents with respect to their severity is shown by Table 9. With such a table, an unambiguous and easy-to-use classification would be possible. However, a lot of research and discussion would be necessary to validate such a table. Similar classification tables are already in use in the car industry, e.g. the new draft ISO 26262 safety standard (2010).

Moreover, however, such a table will never be complete. For accident types not classified in Table 9, either classification could be done by analogy or statistical data or expert judgment could be applied directly. For example, if it were judged that a particular accident typically results in a single fatality, then ID D would be chosen based on Table 5. If statistical evaluation resulted in a FWI value of 0.05 with 90% confidence, then the ID C should be chosen from Table 5.

Finally, the RSM approach should be applied at least to all functions related to interoperability, i.e. that are part of the Technical Specifications for Interoperability (TSIs). For such functions, it may be the case that the THR will have to be fixed within the TSIs to ensure interoperability. In such cases, the complete risk needs to be managed and it needs to be made sure that additional barriers are not assumed that may not exist in all cases. This means that, when defining THRs for technical systems within the TSIs, it needs to be controlled that any additional barriers which are assumed when setting the THR can actually be imposed (for example through other TSIs or other legislation) or assumed present in all circumstances across Europe. This analysis is important in order to control the whole risk to a sufficient level in every case.

CONCLUSION

The risk acceptance and setting of THRs for technical systems can be based on a risk score matrix as explained in this document taking into account a set of typical barriers. This approach is compliant with EC regulations as well as with requirements of the relevant standards.

When using the new Risk Score Matrix approach, mutual recognition will also depend on the list of functions to which the risk matrix is applied. So, the use of a common risk score matrix will facilitate the mutual recognition process, but not lead to an automatic approval.

REFERENCES

Bepperling, S. (2008). *Validation of a semi-quantitative approach for risk assessement on railways* (in German), PhD thesis, Technical University of Brunswick

Bowles, J. (2003) *An Assessment of RPN Prioritization in a Failure Modes Effects and Criticality Analysis*, Proc. RAMS2003, Tampa, 2003

Braband, J. (2005). *Risk analyses in railway automation*. Hamburg: Eurailpress. (in German)

Braband, J. (2011). On the Justification of a Risk Matrix for Technical Systems in European Railways. In Schnieder, E. (Ed.), *FORMS/FORMAT 2010* (pp. 237–288). Springer. doi:10.1007/978-3-642-14261-1_19

CENELEC (1997) E*N 50126 Railway applications –The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*

CENELEC (2010) prEN 15380 Part 4: *Railway applications – Classification system for rail vehicles – Function groups*

DIN (2011) *Semi-quantitative processes for risk analysis of technical functions in railway* signalling (in German), DIN V VDE V 0831-101

EC (2009) *Regulation No. 352/2009 of 24 April 2009 on the adoption of a common safety method on risk evaluation and assessment* as referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council

Hinzen, A. (1993): *The influence of human factors on railway safety* (in German), PhD thesis, RWTH Aachen, 1993

ISO (2010) DIS 26262: *Road vehicles – Functional safety*

Milius, B. (2010). *Construction of a semi-quantitative risk graph* (in German), PhD thesis, Technical University of Brunswick

Sträter, O. (1997) *Evaluation of Human Reliability on the Basis of Operational Experience*, PhD thesis, Technical University of Munich

KEY TERMS AND DEFINITIONS

Barrier: Physical and / or non-physical means planned to prevent, control, or mitigate undesired events or accidents

Common Safety Methods: They describe how safety levels, the achievement of safety targets and compliance with other safety requirements are assessed

Function: A specific purpose or objective to be accomplished that can be specified or described without reference to the physical means of achieving it

Semi-Quantitative: To measure or estimate a parameter by particular classes or intervals. Semi-quantitative methods use ordinal scaled values for parameters, which are combined by mapping (graph, formula or table) into a final result

Technical System: A product developed by a supplier including its design, implementation, and support documentation

Chapter 4
The ForMoSA Approach to Qualitative and Quantitative Model-Based Safety Analysis

Axel Habermaier
Universität Augsburg, Institut für Informatik, Germany

Frank Ortmeier
Otto-von-Guericke University of Magdeburg, Germany

Matthias Güdemann
Otto-von-Guericke University of Magdeburg, Germany

Wolfgang Reif
Universität Augsburg, Institut für Informatik, Germany

Gerhard Schellhorn
Universität Augsburg, Institut für Informatik, Germany

ABSTRACT

This chapter presents ForMoSA (FORmal MOdels and Safety Analysis), an integrated approach for the safety assessment of safety-critical embedded systems. The approach brings together the best of engineering practice, formal methods, and mathematics: traditional safety analysis, temporal logics and verification, as well as statistics and optimization. These three orthogonal techniques cover three different aspects of safety: fault tolerance, functional correctness, and quantitative analysis. The ForMoSA approach combines these techniques to assess system safety in a structured and formal way. Furthermore, the tight combination of methods from different analysis domains results in mutual benefits. The combined approach yields results which cannot be produced by any single technique on its own. The methodology was applied to several case studies from different industrial domains. One of them is an autonomous control of level crossings using radio-based communication, which is used in this chapter to describe the individual steps of the ForMoSA methodology.

DOI: 10.4018/978-1-4666-1643-1.ch004

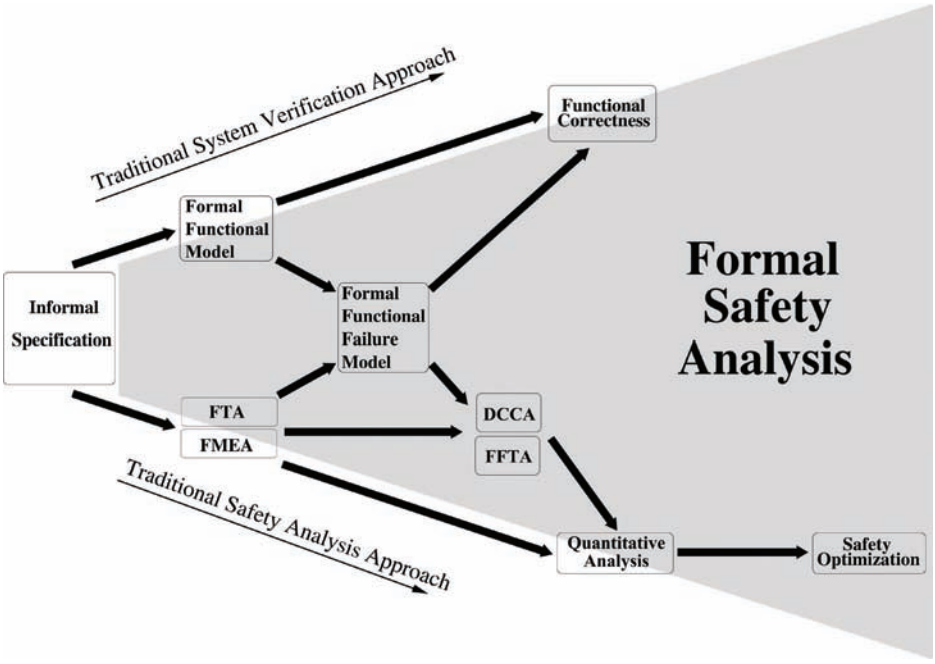
Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

MOTIVATION

Safety-critical systems are expected to operate safely under regular circumstances as well as in many degraded situations. In the latter, these systems have to cope with one or more components that are not working as specified, while at the same time they have to guarantee that no harm is done to people or the environment. Fault Tree Analysis (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002), Failure Modes and Effects Analysis (Reifer, 1979), Hazard and Operability Study (Fenelon, McDermid, Nicholson, & Pumfrey, 1995), and many other traditional safety analysis techniques help safety engineers in systematically analyzing the safety of a system: They dissect the system to determine possible (combinations of) component failures that might result in an occurrence of a hazard and therefore in a violation of the intended behavior. However, the functionality provided by safety-critical systems is becoming increasingly complex, therefore requiring the development of more sophisticated safety analysis techniques to analyze the system behavior under both regular and degraded situations. Additionally, software is becoming a more important factor for the innovation of safety-critical systems; whether it is the automotive sector, avionics, or the railway industry, more and more safety-critical hardware is replaced by software. Among the typical reasons for this change in system design is the increased flexibility offered by implementing certain functionality in software, resulting in a larger amount of features supported by the system as well as more full-fledged implementations of the individual features. On the other hand, software development is complex and error-prone and is thus likely to introduce systematic errors that have the potential of violating safety requirements. In the field of software engineering, formal methods have been developed to deal with software design and implementation errors. These mathematical techniques allow sound reasoning about software designs and implementations with mathematical rigor.

This chapter applies formal methods to the field of safety assessment of safety-critical systems, also pointing out the advantages and disadvantages of the approach over traditional analysis techniques such as the aforementioned Fault Tree Analysis (FTA). The formal techniques presented in the remainder of this chapter focus on discovering cause-consequence relationships under the assumption that all relevant safety requirements, hazards, and component failures are already identified. The techniques also support risk assessment by deriving occurrence probabilities for the hazards from the cause-consequence relationships obtained from the analysis. Costs and other consequences related to the occurrence of a hazard are outside the scope of these techniques, although we briefly outline how the results of the analysis can be used to find an optimal trade-off in the case of antagonistic goals. Furthermore, the techniques presented hereafter can be applied during the entire lifecycle of safety-critical systems, that is, from the beginning of the development of a system until its decommissioning and disposal. During the initial development phase, formal methods are highly recommended by some industry standards such as IEC 61508 (IEC, 2010) and EN 50128 (CENELEC, 2011) in order to reach the highest software safety level. In particular, EN 50128 highly recommends the use of formal methods for the software requirements specification, software design, and verification. Compliance with these standards is often required either by law or contractually. During the maintenance and evolution phase of a system, formal methods can be used to prove that revised safety measures or other changes to the system solve newly identified safety issues or at least do not negatively affect overall system safety. When a system is decommissioned, the techniques can ascertain that a system shutdown does not lead to any safety violations or other unexpected consequences. The techniques also serve as a formal documentation of the safety assessments performed during the entire lifecycle of the system as required by IEC 61508 and EN 50128.

Figure 1. Overview of the ForMoSA approach



In software engineering, formal methods are employed in the synthesis and analysis of software systems. For instance, a mathematical specification of the system requirements improves the likeliness of finding and identifying flaws, misunderstandings, and inconsistencies before the software is actually written, thus reducing the cost to fix these issues. At a later stage of the development process, the implementation can be proven to comply with the formal specification. A formal specification typically lists a set of formal properties that should be satisfied by the system, for example, the program must always terminate or a hazard cannot occur in the presence of certain component failures. However, formal methods cannot fully guarantee adequacy and validity, as it is impossible to decide whether the system that has been developed, modeled, and analyzed is adequate, i.e., whether it fulfills the needs of the people affected by its operation and use. Consequently, the advantage of using formal methods is not the guarantee of correctness, but the increased confidence in the correctness of the final product and the reduced development costs, as more flaws are found in earlier stages of the development process. The downside is the need for skilled engineers trained in the arts of formal methods and the time it takes to formally specify and analyze complex systems. Thus, a trade-off must be found to balance these advantages and disadvantages. For example, it might be a reasonable decision to formally specify only the safety-critical parts of a system and to formally verify only the most safety-critical functions within those parts, whereas the rest of the system is developed in a more informal and traditional way.

OVERVIEW OF THIS CHAPTER

In this chapter, we demonstrate the ForMoSA approach that combines traditional safety analysis, formal methods from software engineering, and statistical methods from mathematics. An overview of the methodology of the approach is shown in Figure 1. It starts with an informal

specification on the left and allows various paths through the diagram, depending on the most important issues. The topmost path – informal specification, formal functional model, and functional correctness – represents the standard approach of software verification in computer science. The bottom one – informal specification, traditional FTA, and quantitative analysis – is the methodology of traditional safety analysis using fault trees. This chapter introduces two techniques, Deductive Cause Consequence Analysis (DCCA) and Formal Fault Tree Analysis (FFTA), which consider safety issues on the basis of formal models and thus yield improved results over traditional safety analysis techniques. The individual steps of the ForMoSA methodology are demonstrated using the case study of a radio-based level control whose informal specification is presented in the next section. In the subsequent sections we introduce the ForMoSA methodology and its application to the case study. In addition, we give a brief introduction to formal methods in general.

Section "Formal System Model" gives a formal specification of the case study using graphical automata specifications. Such a formal functional model specifies the behavior of a system without considering failures and unexpected events. To qualitatively analyze safety, the functional model is extended to a functional failure model that covers all relevant basic failures and failure modes. Manufacturers of standard components often specify relevant failure modes and their probabilities; for instance, relays have the two failure modes "fails to open" and "fails to close" with certain probabilities. Otherwise, failure modes can be derived as the leaves of an informal FTA or FMEA. Section "Formal Failure Models" shows how a given set of failures is systematically integrated into a functional model.

The functional failure model is used to formally analyze safety aspects of the systems by deriving proof obligations that can be checked using the NuSMV model checker. Section "Specification of System Properties" briefly introduces the temporal logic CTL* and explains the formal characterization of the cause-consequence relationships underlying both FFTA and DCCA. FFTA is based on an informal FTA, where the events of the fault tree are specified as CTL* formulae over the formal failure model. FFTA gives a formal meaning to the gates of an FTA by defining proof obligations for each type of gate. These proof obligations are not just Boolean combinations – the typical informal assumption used in FTA –, but allow time to pass between events. We show that no possible causes for the analyzed hazard are omitted if all proof obligations can be verified.

DCCA, on the other hand, analyzes hazards directly, skipping intermediate steps like the construction of a fault tree. It computes minimal critical sets, which in contrast to minimal cut sets obtained from a (formal) FTA are guaranteed to be optimal, as explained in Section "Deductive Cause Consequence Analysis". We also discuss that DCCA is a formalization of both FMEA and informal FTA. However, it is a global technique that provides no insights as to how components of a system contribute to the occurrence of a hazard; it only describes what components contribute. But as both FFTA and DCCA uncover chains of causes and consequences, they can be combined to reap the benefits of both techniques.

Both minimal cut sets and minimal critical sets can be used to estimate hazard probabilities. Many books on safety analysis (Leveson, 1995), (Storey, 1996), (Leveson, 2002) stop at this point, or cover quantitative analysis only marginally. However, the analysis of different design alternatives often has a critical influence on safety. In particular, almost all technical systems have free system parameters such as tolerances, safety margins, or maintenance intervals. These parameters are often chosen by experience; bad choices might only become apparent after an accident has happened. Therefore, Section "Quantitative Analysis and Parameter Optimization" presents an optimization technique that tries to find good parameter values in advance.

At the end of this chapter, we discuss related work and summarize our approach.

A Brief Introduction to Formal Methods

The next three subsections briefly introduce some core concepts in the area of formal methods that are used throughout this chapter. For a more detailed introduction to formal methods in the context of safety-critical systems, the reader is referred to (Haxthausen, 2010) and (Bozzano & Villafiorita, 2010). Readers already familiar with formal methods can safely skip ahead to Subsection "Industrial Use of Formal Methods", which discusses the use of formal methods in industry applications.

Formal Specifications

As in many other engineering disciplines, a specification describes the properties of a product that is to be built or that already exists. As safety-critical systems are typically embedded systems from a software engineering standpoint, a specification of such a system describes both its hardware components and the software controlling its various functions. A formal specification generally describes what a system is supposed to do, as opposed to how it is supposed to achieve its function. Therefore, a specification should neither imply nor favor a specific implementation; the implementation is only required to satisfy (conform to) its specification.

Formal system specifications are typically derived from informally stated requirements. Specifications given in natural language, pseudo code, or diagrams without a formal semantics are considered informal, whereas formal specifications are based on mathematical concepts with precise syntax and semantics, such as grammars defining formal languages. Formal specifications should be seen as a complement to informal specifications: While a formal definition is more precise and less ambiguous, an informal description has the advantage of being more intuitive. Having both an informal and a formal specification combines the best of each approach. Furthermore, formalizing an informal specification also improves the quality of the informal requirements, as possible weaknesses are uncovered and can be fixed accordingly.

Many different kinds of formal specification languages have been developed during the last decades. Which one to use depends on the system that is to be developed. For instance, process algebras such as Hoare's CSP (Hoare, 1985) can be used to describe concurrent systems. Other alternatives include the B method (Abrial, 2007), the term rewriting language Maude (Clavel, et al., 2007), and various kinds of temporal logics; the latter includes CTL* (Baier & Katoen, 2008) which is discussed in more detail in the remainder of this chapter.

Formal Models

In contrast to formal specifications, a formal model of a system describes the structure of the system components and subsystems. It typically also includes a detailed description of the behavior of the components and the interactions taking place between different parts of the system. Abstractions are a mandatory part of any model; i.e., details not deemed to be relevant for the purpose of the model are omitted. For instance, when manufacturers of consumer computers design a new product, they think in terms of components such as the CPU, motherboard, and memory, among others. They do not, however, consider the layout of the transistors within the CPU itself. These details are not relevant for their purpose, but they are of course imperative for the designers of the CPU. Hence, abstractions are crucial to reduce the size of the model. On the other hand, too many abstractions or abstractions in the wrong place might make a model useless or outright inappropriate.

In this chapter, formal models are expressed as program graphs, a special kind of transition system. More commonly known kinds of transi-

tion systems include UML State Machines or Petri nets (Peterson, 1981). The UML (OMG, 2010) is a standard modeling language in the field of software engineering as it defines many different types of diagrams, all of which are tailored towards describing certain aspects of a system and thus allowing the developers to describe the system as a whole. For instance, the aforementioned State Machines describe the dynamic behavior of a system or of one of its components, whereas Component Diagrams can be used to describe the static structure of a system. Sequence Diagrams model the interactions taking place between different components. The UML even includes the specification language OCL that is able to express the requirements a component must fulfill. However, the UML is only semi-formal in the sense that its semantics are not always fully defined. A more detailed discussion about the use of the UML in the context of safety analysis can be found in Section "Related Work".

Formal Verification

The development of a formal model as well as a formal specification of a system typically leads to the discovery of various problems with the informally stated requirements of the system. Even more issues are usually found by using verification techniques that formally prove whether the (model of) the system satisfies the specification. Formal verification is superior to testing and simulation in the sense that it considers all possible system executions, whereas only a small amount of them can be tested or simulated. Still, tests and simulations remain important, as they can be performed on the real system and not only on the model. They can thus detect issues that have been abstracted away during the construction of the formal system model or that cannot be discovered by formal verification due to false assumptions. For example, the verification might assume that a sensor detects critical values within one second, whereas the real sensor can only guarantee detection within a couple of seconds. In this case, the formal model and the real system do not match and the verification is delusive. Extensive tests and simulations of the real system are the only chance to find such discrepancies.

This chapter focuses on model checking to prove that a model satisfies the specified properties. Other techniques include computer assisted interactive theorem proving (Balser, Reif, Schellhorn, Stenzel, & Thums, 2000) which is, however, more time-consuming and more difficult in contrast to model checking. Model checking is a fully automated technique that explores all possible states the system can reach over time. The specification of a system imposes constraints on how the system is allowed to evolve, i.e., which states it is allowed to reach and in which order. A model checker can therefore unequivocally decide whether the specification is satisfied. If it is not, a model checker generates a sequence of states that explains how the system is able to reach an undesired state. This counterexample to the validity of a specified system property is useful in determining the cause of the violation, therefore contributing to the understanding and the fix of the problem. However, model checking is generally limited to finite state systems, that is, for systems which have a finite amount of possible system states. Additionally, the state space of the system should not only be finite, but also as small as possible: Many real-world systems quickly reach a size that precludes the use of model checkers due to computation time and memory constraints. In the literature, this issue is often referred to as the state space explosion problem. The only way to reduce the size of the state space is to use more aggressive abstractions, which, if done wrong, can have a negative impact on the validity of the results for the real-world system as already discussed above.

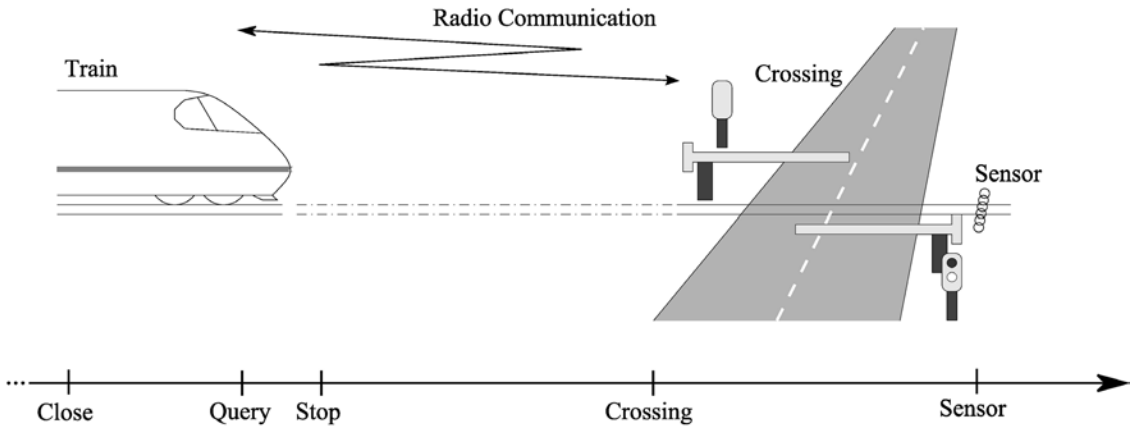
As for the modeling and specification mechanisms, it depends on the system at hand which model checker is suitable. The main difference between different model checkers concerns the type of the models they are able to check. For instance, UPPAAL (Bengtsson & Yi, 2004) supports timed models, whereas PRISM (Kwiatkowska, Norman, & Parker, 2011) additionally considers probabilities of state transitions. SPIN (Ben-Ari, 2008) and NuSMV (Cimatti, et al., 2002), on the other hand, support neither of the aforementioned characteristics, but differ in the algorithms they use to perform the checks as well as their modeling and specification languages. In the following, the symbolic model checker NuSMV is used to verify system properties, which is one of the most efficient tools available.

Industrial Use of Formal Methods

Today, formal methods are not widely in use within the industry despite the possible advantages they offer. However, in the case of safety-critical systems, many international standards like IEC 61508 for safety-critical software development highly recommend the use of formal methods in system design and safety analysis. EN 50128 addresses software development for railway systems and is closely linked to IEC 61508; it also highly recommends the use of formal methods in order to reach the highest safety integrity level, and even explicitly lists some of the aforementioned specification formalisms like CSP and temporal logics. Even the older DO-178B standard for the avionics sector (RTCA, 1992) mentions formal methods, even though they were evaluated to have an inadequate maturity level at that time. However, almost twenty years later, significant advancements have been made in the area of both formal techniques and tools. Due to these improvements and the more strict requirements of the standards, industry projects especially in the transportation sector increasingly rely on the use of formal methods, as indicated by the recently conducted survey of (Woodcock, Larsen, Bicarregui, & Fitzgerald, 2009). The survey also shows that specification and modeling as well as model checking are among the techniques mostly employed today in the industry. In particular, within the last decade the use of model checking has increased from 13% to 51% across all examined industry projects. Additionally, the use of formal methods is reported to have an overall positive effect on development time, cost, and quality. Most notably, the participants of the survey reported no cases of a decrease in quality.

In the railway sector, formal methods were used successfully in the development of the Paris Roissy Airport shuttle (Abrial, 2007). The specification of the safety-critical parts of the system was constructed in B, and the subsequent interactive proofs made it possible to skip unit testing the system and to perform global integration tests immediately. All those tests were successful, resulting in a significant overall cost reduction. Another success story of the use of formal methods in large-scale real-world applications is NASA's Deep Space 1 mission (Havelund, Lowry, Pecheur, Penix, Visser, & White, 2000): With the help of the SPIN model checker, five concurrency errors were found in the LISP code of the Remote Agent software before launch in 1997. The developers affirmed that they were very unlikely to identify these problems through tests. After Deep Space 1 was launched in 1999, a deadlock occurred in another component that had not been model checked and remained undetected despite over 300 hours of system testing performed by NASA engineers. Using SPIN again, the problem was quickly identified to be the same as one of those concurrency errors identified in 1997. Further examples of the application of formal methods in industry applications are listed in (Woodcock, Larsen, Bicarregui, & Fitzgerald, 2009) and (Bozzano & Villafiorita, 2010). The latter also discuss some further aspects of safety standards and certification with regard to the use of formal methods.

Figure 2. The approaching train queries the crossing to ascertain that it can safely pass



OVERVIEW OF THE CASE STUDY

The case study presented throughout this chapter was the reference case study of the Deutsche Forschungsgemeinschaft (DFG) project 1064 "Integration of Software Specification Techniques for Applications in Engineering". It is based on a novel technique to control railroad crossings on medium speed routes, i.e., routes with a maximum speed of 160 km/h, suggested by the German railway organization Deutsche Bahn. Compared to the original specification there are some simplifications that reduce its scope and remove some complexities. With this we intend to help the reader focus on the techniques presented below and their application to the case study, while omitting some of the complexities of the real-world system. A more detailed specification of the case study including the description of a demonstrator can be found in (Hänsel, Poliak, Slovák, & Schnieder, 2004). Section "Related Work" lists some publications which analyze the more detailed version of the case study.

In the proposed control system for railroad crossings, many of the sensors and signals traditionally placed alongside the tracks are replaced by radio communication and software computations, thereby effectively decentralizing the system and making it more robust and scalable overall. The train control software knows about the locations of the crossings that lie ahead and secures them just in time via wireless communication. Among other problems, the system takes lost communication messages and malfunctions of the crossing into account, causing the train to initiate an emergency stop before reaching the crossing if it cannot be certain that the barriers are in fact closed. Consequently, safety-critical functionality moves from hardware components to software-based control systems that are specifically designed to avoid collisions, i.e., situations in which a train passes an unsecured crossing.

Figure 2 shows a train approaching a crossing. In this chapter, we only consider a single-way track with the train always approaching "from the left". The five positions highlighted at the bottom of the figure indicate the locations of the crossing and the sensor along the track as well as the *Close*, *Query*, and *Stop* positions that are computed dynamically by the control software. When the train reaches the *Close* position, it sends a message to the crossing requesting it to initiate its closing procedure. This procedure takes some time, i.e., the crossing does not immediately enter a secured state as it must first activate its traffic lights for some time before the barriers are allowed to begin closing, which in turn also takes some time. Once the barriers are closed,

the crossing is secured and the train is allowed to pass. During the time the barriers are closing, the train continues to approach the crossing. Eventually, it reaches the *Query* position, prompting the control software to query the crossing's state. The crossing responds with a message indicating that it is secured and that the train can safely pass. If, however, the train does not receive the message before it reaches the *Stop* position, it initiates an emergency stop as it does not know whether it is safe to pass the crossing. There might have been a malfunction that prevented the barriers from closing like a motor defect or the loss of the train's closing message, so the train cannot be allowed to pass the crossing. On the other hand, it is possible that there is merely a communication malfunction resulting in the loss of the crossing's message while the barriers are in fact closed. In that case, the emergency stop is unnecessary, but from the train's point of view there is no way to tell these situations apart. It must always stop, regardless of the reasons that cause the crossing's message not to arrive in time.

In order to compute the *Close*, *Query*, and *Stop* positions, the train uses data about its position and maximum deceleration as well as data about the parameters of the crossing it is about to pass. The train accesses a data store that holds the relevant parameters of all railroad crossings, including their positions and the time they need to secure themselves. Additionally, an odometer continuously measures the train's current speed and position. The *Stop* position is constantly updated such that the train is always able to fully stop in front of the crossing. The position depends on the train's current speed, its distance to the crossing, and its maximum deceleration. Additionally, the calculations take appropriate safety margins into account in order to compensate for small discrepancies in the data returned by the odometer. The *Query* position depends on the *Stop* position: The time it takes the train to travel the distance between both positions must be greater than the time it takes to query the crossing's state. Similarly, the *Close* position depends on the *Query* position, as the train must allow the crossing to secure itself before it queries the barriers' state. Messages sent between the train and the crossing can be delayed by a certain amount of time, which has also to be taken into account when computing these positions.

There is a sensor behind the crossing that signals the crossing to open once the approaching train has passed by. The sensor and the crossing are connected by wire. The barriers remain closed until the approaching train passes the sensor unless the train takes too long to reach the crossing, in which case the barriers open regardless of the train's position. While this is seemingly counterproductive at first glance with regard to collision avoidance, it does indeed make sense according to the statistics: The longer the barriers are closed, the more likely it is that some drivers ignore them, i.e., drive around them and pass the secured crossing. This seems even more likely for pedestrians. At least the drivers can easily be prevented from doing so by using barriers that cover the entire width of the crossing (in contrast to the barriers shown in Figure 2). Due to this human factor, it is better to open the barriers after a certain period of time rather than increasing the likelihood of this uncontrollable situation.

Collision avoidance is the system's primary safety goal, i.e., the system is designed to avoid the hazard of a train passing an unsecured crossing. A hazard concerning the erratic human behavior mentioned above occurs if the barriers remain closed for longer than absolutely necessary. These two hazards are obviously antagonistic, as the former can be avoided by never opening the barriers while the latter is made impossible by never closing them. However, we only consider the first hazard in this chapter but give an outlook on how to find an optimal solution for the case of such contradictory hazards in Section "Quantitative Analysis and Parameter Optimization".

In the remainder of this chapter, we construct a formal model of the case study that describes the

system's behavior precisely, that is, mathematically, whereas the informal description found in this section is prone to accidental omissions of critical details and generally leaves room for interpretation, possibly causing misunderstandings. During the formalization of the system, we make some simplifying assumptions to decrease the complexity of the case study; e.g., we do not model the traffic lights at the crossing and we assume that the train reliably knows the position of the next crossing it is going to pass. We introduce two formal safety analysis techniques and apply them to the case study to check whether the system prevents the aforementioned hazard in the absence of and with the occurrence of component failures. We analyze precisely which combinations of component failures lead to the hazard. Along the way we also check that the system is functionally correct in the sense that the hazard does not occur in the absence of any failures.

Formal System Model

Reasoning about the properties of a system that is only informally specified is often imprecise or even impossible. There are typically many cases where natural language allows different interpretations of a statement, leading to ambiguities and misunderstandings. Moreover, once a certain level of complexity is reached, no single person is able to fully understand all the different facets and behaviors of the system. A formal system model, on the other hand, opens up the possibility of mathematical proofs of system properties that can be performed either by hand, semi-automated with the help of a theorem prover like KIV (Balser, Reif, Schellhorn, Stenzel, & Thums, 2000), or fully automated using a model checker like NuSMV (Cimatti, et al., 2002). We focus on enabling fully automated verification using NuSMV in this chapter, even though our tool KIV supports one of the analysis techniques presented below, namely Formal Fault Tree Analysis. But as working with KIV is typically hard and time-consuming, we generally prefer model checking whenever possible. However, our analysis techniques are not in any way tied to KIV or NuSMV and can be used in conjunction with many different specification formalisms and tools.

Formal methods are traditionally used to verify system correctness. The informal description of the system behavior is formalized by means of specification formalisms like algebraic specification or some kind of transition system. The properties – e.g., the observable behaviors or guarantees – that the system should expose are formalized over the formal model. Specification languages for properties include first-order logic or temporal logics, among many others as mentioned in Section "A Brief Introduction to Formal Methods". Whether a specific formalization mechanism is suitable for a given system and its properties is influenced by many factors, like the tools that are used for verification or code generation, the complexity of the system, the finiteness or infiniteness of the system, or temporal relationships that are of importance.

In this chapter, we use formal methods to check not only functional correctness but also safety relevant properties of the system under consideration. Functional correctness is a prerequisite for safety analysis, because it is the most basic safety property. If the system is not functionally correct, safety is typically violated anyway. In fact, the safety analysis techniques presented below also guarantee functional correctness with respect to the analyzed hazards.

All formal methods have one common weakness: The results of the verification are only as good as the underlying system model. If the model does not match the real system, the results obtained from the formal analysis do not necessarily reflect the system's actual properties. In a sense, formal safety analysis is even harder than correctness verification because the system's behavior in the case of failures must be considered and modeled in addition to the desired functional behavior. We deal with modeling erroneous behavior in Section

"Formal Failure Models" and focus exclusively on the functional behavior for the moment.

The semantics of the temporal logics in which we express the system properties are based on Kripke structures, introduced in the next subsection. The safety analysis techniques presented in this chapter are independent of the actual specification language used to describe the system, provided that the specification language can be mapped to Kripke structures. Subsection "Modeling Language" presents the modeling language that we use in this chapter and also defines its mapping to Kripke structures. Subsequently, Subsection "Application to the Case Study" presents the formal model of the case study's functional behavior, disregarding all possible component failures for the moment. It also highlights some of the difficulties one often has to deal with when formalizing an informal system specification.

Kripke Structures

Transition systems are a standard class of formal models that represent the states and the behavior of hardware and software systems. They can be defined in many different ways ranging from very simple ones (like the Kripke structures defined below) to very complex ones (like UML State Machines) with sophisticated semantics. The latter typically expose more expressive modeling features that increase the understandability of the system models. On the other hand, verification of such models becomes more expensive as the semantics of these features are typically quite complex as demonstrated by the run-to-completion semantics of UML State Machines (Knapp, 2004). In this chapter we therefore try to strike a balance between the expressiveness of the syntactic constructs and the straightforwardness of the semantics. To this end, program graphs are defined in the next subsection. First, however, Kripke structures are formally defined as they form the basis of our safety analysis techniques. Many different specification languages and flavors of transition systems can be mapped to semantically equivalent Kripke structures; the next subsection presents such a mapping for program graphs. Hence, our techniques are not restricted to a particular kind of specification language; rather they just assume that the semantics of the specification can be mapped to the following definition of Kripke structures:

Definition 1: (Kripke Structure) A Kripke structure K is represented by a tuple (AP, S, R, L, I) where

- AP is the set of atomic propositions,
- S is the (finite) set of states,
- $R \subseteq S \times S$ is the (left-total) transition relation,
- $L : S \to 2^{AP}$ is the labeling function that defines for each state $s \in S$ the set $L(s)$ of all atomic propositions that are valid in s,
- $I \subseteq S$ is the set of initial states

Together with the set of initial states the transition relation defines the set of paths of a Kripke structure:

Definition 2: (Paths of a Kripke Structure) Let $K = (AP, S, R, L, I)$ be a Kripke structure. An infinite sequence $\sigma = s_0, s_1, \ldots$ of states with $s_i \in S$ is a path of K if $s_0 \in I$ and $(s_{i-1}, s_i) \in R$ for all $i > 0$. $paths(K)$ denotes the set of all paths of K.

Generally, $paths(K)$ is represented by an infinite tree, as any state of K is allowed to have more than one successor state. It is infinite because of the left-totality of the transition relation which ensures that every state has at least one successor state.

Modeling Language

The following definition of program graphs is based on the one found in (Baier & Katoen, 2008).

In the remainder of the chapter, the case study is modeled using program graphs. As already mentioned, other kinds of specification languages like Harel Statecharts (Harel & Naamad, 1996), UML State Machines, or SCADE (Abdulla, Deneux, Stålmarck, Ågren, & Åkerlund, 2006) could have been used too (Section "Related Work" contains an overview of some other specification languages that have been used to model the case study). All of these formalisms can be mapped to equivalent Kripke structures, as we show for the case of program graphs below. Program graphs as presented here are geared towards being easily convertible into the NuSMV input language.

Definition of Program Graphs

Program graphs depend on a set of atomic propositions AP_V over a set V of typed variables with finite ranges. We assume that propositions $p \in AP_V$ are of the form $v = c$, i.e., a proposition always compares a variable v to a constant c that lies within its range. Furthermore, F_V denotes a propositionally closed language over AP_V, which is able to express comparisons between different variables in V by reduction to atomic propositions. For instance, the formula $x = y$ can be reduced to $(x = 0 \wedge y = 0) \vee (x = 1 \wedge y = 1) \vee ... \vee (x = n \wedge y = n)$ where the range of x and y is $[0, n]$. A variable environment $\eta \in Env_V$ assigns a value of the correct type to each variable in V, always respecting the variables' ranges. We write $\eta \models \phi$ to mean that η satisfies $\phi \in F_V$. For example, $\phi = (x \geq 4 \wedge y = A)$ is a formula in F_V and $\eta \models \phi$ holds for $\eta(x) = 17$ and $\eta(y) = A$, for example. The validity of any formula $\phi \in F_V$ is decidable because we consider finite variable ranges only.

Moreover, we assume a language A_V of actions for manipulating variable environments. An action α is either empty, i.e., $\alpha = \varepsilon$, an assignment of an expression to a variable, that is, $\alpha = (x := e)$, or a parallel combination of actions $\alpha_1 \mid \alpha_2$. That is, an action is a simultaneous assignment of a multitude of variables. We assume that expressions allow basic arithmetic on variables in V. We write $(\eta, \eta') \models \alpha$ to mean that $\alpha \in A_V$ transforms the variable environment η into the environment η'; note that such a transformation and the corresponding action must be well-formed. A transformation is well-formed if it respects the ranges of the variables written by $\alpha,$, i.e., if it does not try to assign a value outside the range of any written variable. An action is well-formed if it does not contain conflicting parallel actions like $x := 1 \mid x := 2$. Transformations and actions that are not well-formed are forbidden.

With these preliminaries, our notion of program graphs can be defined as follows:

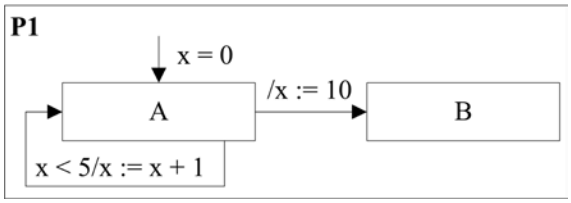
Definition 3: (Program Graph) A program graph $P : (V, S, \rightarrow, s_0, \phi_0)$ is defined such that

- P is the name of the program graph,
- V is the set of variables,
- S is the (finite) set of states,
- $\rightarrow \subseteq S \times F_V \times A_V \times S$ is the transition relation,
- $s_0 \in S$ is the initial state,
- $\phi_0 \in F_V$ is the formula that holds initially, thereby determining the initial variable environment.

We write $s_1 \xrightarrow{\phi/\alpha} s_2$ for a transition $(s_1, \phi, \alpha, s_2) \in \rightarrow$ with s_1 denoting the original state and s_2 denoting the new one. If the condition ϕ – also called guard – trivially holds, that is, $\phi \Leftrightarrow true$, we write $s_1 \xrightarrow{/\alpha} s_2$. Conversely, $s_1 \xrightarrow{\phi} s_2$ denotes a transition without any action, i.e., $\alpha = \varepsilon$, and $s_1 \rightarrow s_2$ means that $\phi \Leftrightarrow true$ and $\alpha = \varepsilon$.

Program graph P1 in Figure 3 consists of the variable x and two states A and B, the former being the initial state. Let $[0,10]$ be the range of

Figure 3. A simple program graph



x, with 0 being the initial value. In state A, the behavior of P1 depends on the current variable environment η: If $\eta \models x < 5$, one of the transitions is chosen nondeterministically. So either P1 switches to state B and because of $(\eta, \eta') \models (x := 10)$, it follows that $\eta'(x) = 10$, or it remains in state A and x is incremented by one. Otherwise, the transition to state B is taken deterministically.

The transition relation $\rightarrow$ is not required to be left-total. Thus, a program graph can reach a state where no outgoing transition exists (see state B in Figure 3), none of the guards of the outgoing transitions currently hold, or all actions of all outgoing transitions violate the ranges of the assigned variables. If a program graph ever encounters such a situation where it were stuck, an implicit stuttering transition would be taken that re-enters the current state and has action $\alpha = \varepsilon,$. This implicit transition is taken deterministically, that is, it is taken only if no other transition is possible. In the following, we only consider such program graphs with infinite paths, i.e., we assume that a program graph can always take some transition in any state at any time.

A program graph can be mapped to a Kripke structure as follows. This mapping enables us to apply our safety analysis techniques, which are defined over Kripke structures, to all systems modeled as program graphs.

Definition 4: (Kripke Structure Semantics of a Program Graph) A Kripke structure $K_P = (AP, S, R, L, I)$ corresponds to a program graph $P : (V, S', \rightarrow, s_0, \phi_0)$ with infinite paths only if

- $AP = AP_V \cup S'$,
- $S = S' \times Env_V$,
- $R = \{((s_1, \eta_1), (s_2, \eta_2)) \mid s_1 \xrightarrow{\phi/\alpha} s_2 \wedge \eta_1 \models \phi \wedge (\eta_1, \eta_2) \models \alpha\}$
- $L\big((s, \eta)\big) = \big\{ p \in AP \mid \eta \models p \big\} \cup \{s\}$,
- $I = \big\{ (s_0, \eta) \mid \eta \models \phi_0 \big\}$.

In particular, the totality of the transition relation R of the Kripke structure is guaranteed because we consider program graphs with infinite paths only.

Synchronous Composition of Program Graphs

Typically, each system component is modeled by its own program graph to increase the modularity of the model. Besides increasing the readability, another advantage of the modular approach is the possibility to reuse models to represent instances of several different system components of the same kind. For example, in the formal model of the case study there is one program graph that models the behavior of the crossing, a second one that represents the train, a generic delay timer used for all timeouts and communication delays, and so on. These program graphs operate on shared state, as components typically depend on other components. Therefore, it is necessary to formally define how several individual program graphs can be composed together to obtain the system as a whole.

Synchronous composition of program graphs is defined below. Synchronously combined program graphs operate in lockstep, i.e., all individual program graphs perform one transition during each step of the system. We can adequately model the case study using synchronous composition which is also generally preferred by NuSMV. It is also possible to use asynchronous parallelism where

execution is interleaved, that is, only one of the program graphs performs a step at a time.

Definition 5: (Synchronous Composition) Let $P_i : (V_i, S_i, \rightarrow_i, s_{0,i}, \phi_{0,i})$ for $i = \{1,2\}$ be two program graphs. The synchronous composition $P_1 \,||\, P_2$ is given by

$$P_1 \,||\, P_2 : (V, S_1 \times S_2, \rightarrow, (s_{0,1}, s_{0,2}), \phi)$$

where

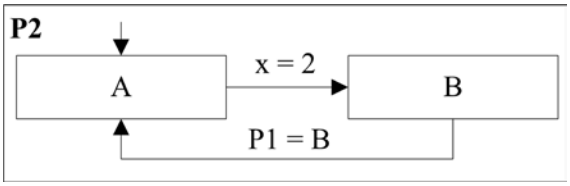
- $V = V_1 \cup V_2 \cup \{P_1 : S_1, P_2 : S_2\}$
- $\phi \Leftrightarrow \phi_{0,1} \wedge \phi_{0,2} \wedge P_1 = s_{0,1} \wedge P_2 = s_{0,2}$
- the transition relation $\rightarrow$ is defined by the rule

$$\frac{s_1 \xrightarrow{\phi_1/\alpha_1}_1 s_1' \qquad s_2 \xrightarrow{\phi_2/\alpha_2}_2 s_2'}{(s_1, s_2) \xrightarrow{\phi_1 \wedge \phi_2 / \alpha_1 | \alpha_2 | P_1 = s_1' | P_2 = s_2'} (s_1', s_2')}$$

The set of variables of the composed program graph is the union of the original variable sets; therefore, variables are shared by name. If the two program graphs define a variable with the same name but different ranges, the synchronous composition is undefined. Additionally, we add two new state variables P_1 and P_2, which range over the set of states of their respective program graph P_1 or P_2. The initial condition and the transition relation of the synchronous product guarantee that these state variables are always set to the correct value, that is, to the value representing the current state of their respective program graph. In particular, no other program graph can assign to these state variables, as that would create conflicting actions.

Note that in the case of conflicting actions resulting from the synchronous composition of two program graphs, $\alpha_1 \mid \alpha_2$ is not well-formed and therefore forbidden. In that case, the rule above cannot be applied and the transition does not exist. As a consequence, the combined program graph might not have all transitions one might expect it to have. However, this problem can be avoided by following a simple design rule: Every variable may only be written by exactly one program graph; obviously, no conflicting actions can occur if this rule is followed. Note that two program graphs with only infinite paths result in a synchronous product with only infinite paths. Due to the implicit stuttering behavior that we assume for program graphs, this observation even holds if transitions are lost because of conflicting actions.

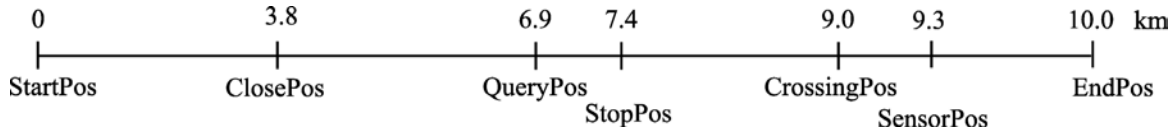
Figure 4. Another simple program graph



In a synchronous composition of two program graphs, shared variables are identified by name. In Figure 2, for example, program graph P2 is defined with variables x and $P1$. When it is synchronously combined with program graph P1 shown in Figure 3, P2 leaves its initial state A once P1 has set the value of x to 2 and returns to state A once P1 is in state B. The variable $P1$ defined in P2 is automatically combined with the implicit state variable $P1$ denoting P1's current state. Variables must be renamed to avoid unintended sharing. If, for instance, the variable x of P2 should be distinct from variable x of P1, all occurrences of x must be replaced with some other variable name in either P1 or P2.

The paths of the Kripke structure corresponding to a synchronously composed product P1||P2 of two program graphs P1 and P2 differ from the paths of the product P2||P1 since the states of both product automata are structured differently. To overcome this problem, we abstract from the concrete states of a program graph (see Figure 4) and instead look at the sequences of atomic propositions that hold in each state of a path of the

Figure 5. Positions in the formal model of the case study



corresponding Kripke structure. Section "Formal Failure Models" depends on these so-called traces of program graphs.

Definition 6: (Traces of a Program Graph) Let P be a program graph and $K_P = (AP, S, R, L, I)$ be its corresponding Kripke structure. The set $traces(P)$ of traces of P is defined as:

$$traces(P) = \{L(s_0), L(s_1), ... \mid s_0, s_1, ... \in paths(K_P)\}$$

It is possible to reconstruct the original state as well as the original variable environment of P from each set of atomic propositions $L(s)$ (see Definition 4).

Application to the Case Study

This section presents a formal model of the railroad crossing case study using the graphical notation of program graphs. As we intend to check the system's safety properties using fully automated model checking, the number of possible system states must be finite; in particular, the formal model must not only be state finite, the number of states should also be as low as possible for performance reasons. Parallel composition typically causes the number of states to grow exponentially, which poses a significant problem for the verification of large systems using model checking.

The position and the speed of the train both play an important role in the case study, both of which are non-discrete real functions over time and are therefore not suited for model checking with NuSMV. Hybrid model checkers exist that support continuous functions directly, thus we consider hybrid model checking for future work. For now, we express the system in a discrete and finite way and convert system constants like the train's maximum speed accordingly.

Discrete Values, Finiteness, and System Constants

As we are only interested in analyzing the possible causes of a collision, the position of the train is only of importance as long as the train is in the proximity of the crossing. Hence, we are able to restrict the positions of the train to an interval of fixed length beginning at some point in front of the *Close* position and ending behind the position of the sensor. The entire interval and the relevant positions in between are shown in Figure 3.

The train is allowed to remain at position 0 km for an indefinite amount of time, meaning that it is approaching the crossing but is still too far away to affect the system at all. Eventually, it leaves the start position and "enters the system", i.e., it is closing in on the *Close* position and is now of relevance to the system (see Figure 5). The crossing and the sensor are located 9.0 km and 9.3 km away from the start position, respectively. Once the train has passed the sensor, it can no longer cause a collision on this crossing – and we assume that if the system is safe for one crossing, it is safe for sequences of crossings – so we stop tracking its exact position after 10.0 km. We model the system at a resolution of 3 m; hence there are 3333 distinct positions. As a result, the position of the crossing is 3000 in our model, whereas the sensor is positioned at 3100.

Time is modeled at a resolution of two seconds per system step. The train's maximum speed is

set to 29 in the model which corresponds to a real-world speed of 29 * 3 m / 2 s = 43.5 m/s = 156.6 km/h, slightly below the maximum speed of 160 km/h. Consequently, there are 29 different speed values, the delta of 1 corresponding to a speed change of roughly 5.4 km/h. We assume a deceleration of ¾ m/s^2 – equivalent to a value of 1 in the model –, causing the train to stop after about 1.3 km. The precision could be improved by using higher resolutions for time and position at the expense of the time and memory it takes to perform the model checking. The values chosen for this case study are sufficiently precise and allow the model to be checked within a few minutes on a standard desktop computer using NuSMV.

We choose a communication delay of two seconds per message, and set the amount of time the crossing needs to secure itself to one minute. The timeout that opens the barriers regardless of the train having passed is set to four minutes.

All of the aforementioned values are constants in our system model. If chosen unwisely, they can negatively affect the result of the safety analysis; therefore, it is best if they can be obtained from the informal specification of the system. For example, if the timeout is set to a value such that the train has no chance of ever reaching the barriers before they open, the system might either be unsafe or the train always stops in front of the crossing. Thus care must be taken when choosing these constants and it is generally a good idea to perform some sanity checks before further analyzing the system. To prevent the aforementioned problem, it should be checked whether the system indeed allows the train to pass the crossing safely or whether it reaches the end position at all. Furthermore, the system should never reach a state where the speed of the train is 0 – as we do not yet model any component failures, the train should always safely pass the crossing. This cannot be proven, for instance, if the barrier timeout is set too short, i.e., the barriers are open again before the train even reaches the *Query* position. If the system does not pass these sanity checks, it might be proven to be safe because no safety critical state is ever reached. Obviously, the system does not model the intended functionality at all and the results of the analysis are not only worthless but plain wrong.

The formulae to compute the *Close*, *Query*, and *Stop* positions are derived from the equations of motion. The *Stop* position depends on a safety margin of 350m: 92m account for discrepancies in the readings of the odometer while the remaining 258m are required for technical reasons of which 45m account for rounding errors, 39m are needed due to the effects of the discrete position modeling, and 174m compensate for the delays between reaching the *Stop* position and initiating the emergency stop as explained in the next subsection; this delay is even higher for the *Close* and *Query* position, so these also depend on a safety margin. Simply by coincidence, the safety margins are the same for the calculations of all three positions, albeit for different reasons.

$$
\begin{aligned}
StopPos &= CrossingPos - Speed^2 \\
&\quad /(2 * Deceleration) - SafetyMargin \\
&\quad = 7.4\text{km} = 2464 \\
QueryPos &= StopPos - 2 * CommDelay * Speed \\
&\quad -SafetyMargin = 6.9\text{km} = 2290 \\
ClosePos &= QueryPos - (CommDelay \\
&\quad +ClosingDelay) * Speed - SafetyMargin \\
&\quad = 3.8\text{km} = 1275
\end{aligned}
$$

Formal Model of the Case Study

The formal model of the case study consists of several synchronously composed program graphs representing the following system components: the crossing, the train control software, the brakes, the train's position and speed measurements, and several timeout and communication automata. These program graphs, presented throughout the remainder of this section, are synchronously composed to form the overall formal system model.

Figure 6. The program graph modeling the train's current position

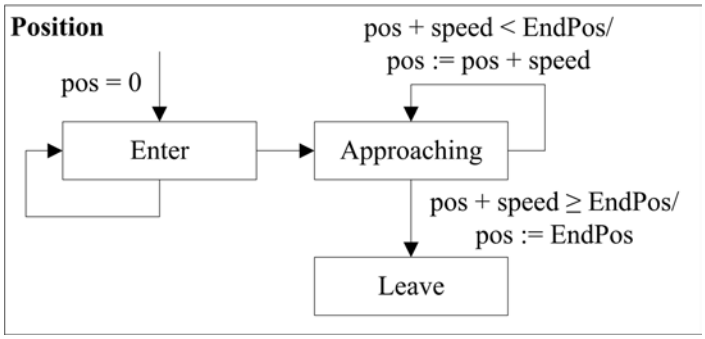
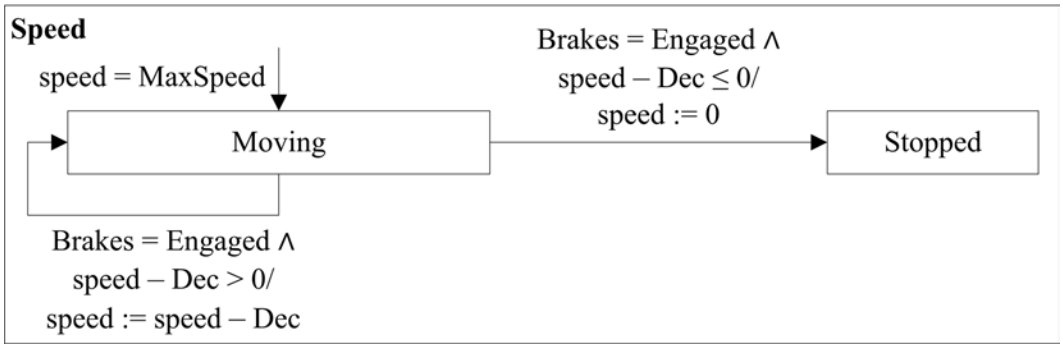


Figure 7. The program graph modeling the train's current speed

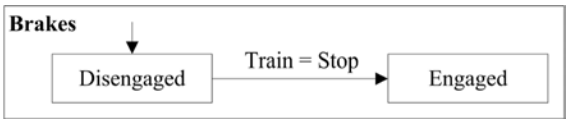


The Position program graph in Figure 6 models the position of the train. Initially, the position is 0, which it can remain for an arbitrary amount of time. Eventually, the train starts approaching the crossing and enters the *Approaching* state, where its position is updated depending on the train's current speed. Should the train initiate an emergency stop, the speed eventually becomes 0 and the Position program graph remains in state *Approaching*. Otherwise, it enters the *Leave* state once the position reaches the end of the position interval.

Figure 7 shows the model of the train's speed. It initially moves with maximum speed, only entering state *Stopped* once the speed reaches 0. Note that it is generally not possible to subtract the deceleration from the current speed, as the units of measure differ. However, in the model the variable Dec implicitly takes the elapsed time into account. The same applies to the position and the speed variables of the Position program graph.

Our model does not allow the train to accelerate or decelerate at will, rather it either moves at maximum speed or the brakes are engaged and the train is stopping. Arbitrary acceleration and deceleration is not considered for the case study, although it is of course possible to do so. However, a model in which the train is allowed to change its speed at will introduces a huge amount of additional states, significantly increasing the time the model checker needs to verify the system properties. On the other hand, it is obviously a single point of failure: If the train is delayed for too long after passing the *Stop* position – e.g., it stops for some time and then continues approaching – it assumes the crossing to be safe whereas the barriers are already open again when it finally reaches the crossing. (Hänsel, Poliak, Slovák,

Figure 8. The program graph modeling the train's brakes



& Schnieder, 2004) explain how this issue is resolved in the full version of the case study.

The behavior of the brakes is shown in Figure 8. The brakes are disengaged for as long as the train control software does not enter the *Stop* state as explained below. Once engaged, the brakes remain engaged, causing the train to come to a full halt. We do not allow the train to accelerate again.

The control software is modeled in Figure 9. The software remains idle until the train reaches the *Close* position. Due to the discretization of the position, the train may never reach the *Close* position exactly, but "jump" over it, so $pos = ClosePos$ would not be an adequate formalization of the condition. The software remains in the *Wait* state until it is time to query the crossing. In the *Query* state, the control software waits until either the secured message is received or the stop position is reached, switching to the *Go* or *Stop* state accordingly.

As mentioned above, there is a safety margin that compensates for the delays between reaching the *Close*, *Query*, and *Stop* positions and initiating the corresponding action. When the train control software switches to the *Stop* state, the brakes do not immediately switch to state *Engaged*. There is a delay of one step before the brakes are engaged and another delay of one step before the speed actually starts reducing. Thus, after reaching the *Stop* position, the train continues to travel at full speed for the next two time steps. Similar arguments can be made when the train reaches the *Close* and *Wait* positions. However, the overall system behavior is not influenced by this delay, provided that the safety margin is adjusted accordingly.

These delays are a general problem with program graphs and other kinds of transition systems; one possible solution is to define micro/macro step semantics similar to Statecharts (Damm, Josko, Hungar, & Pnueli, 1998) where an arbitrary but finite amount of micro steps occurs within each macro step. A macro step corresponds to a system step that takes some time. Micro steps, on the other hand, do not take any time and allow the system to stabilize, for example, by seemingly immediately reacting to evens like reaching the *Stop* position. Micro/macro step semantics introduce additional states, restricting the size of the models that can be efficiently model checked. Even though it is possible to extend program graphs with micro/macro steps semantics, we refrain from doing so in the interest of not further complicating the discussion of the case study.

The train and the crossing exchange communication messages. Program graphs and NuSMV, however, do not support message passing directly, although there are other types of transition systems and model checkers that do so. We therefore model the reception of a message indirectly by checking whether the corresponding communication automaton is in the Signal state. The communication automata result from instantiating the program graph in Figure 8 with the values listed in Table 1; the parameters of the program graph are the activation condition $\phi,$, the maximum counter value $Delay$, and its name. The close and query messages sent to the crossing are not visible in the program graph in Figure 9; rather, the sending of a message is modeled by the activation condition ϕ of the corresponding communication automaton. There are three different messages sent between the train and the crossing, corresponding to the CommClose, CommQuery, and CommSecured automata, respectively: the mes-

Figure 9. The program graph modeling the train's control software

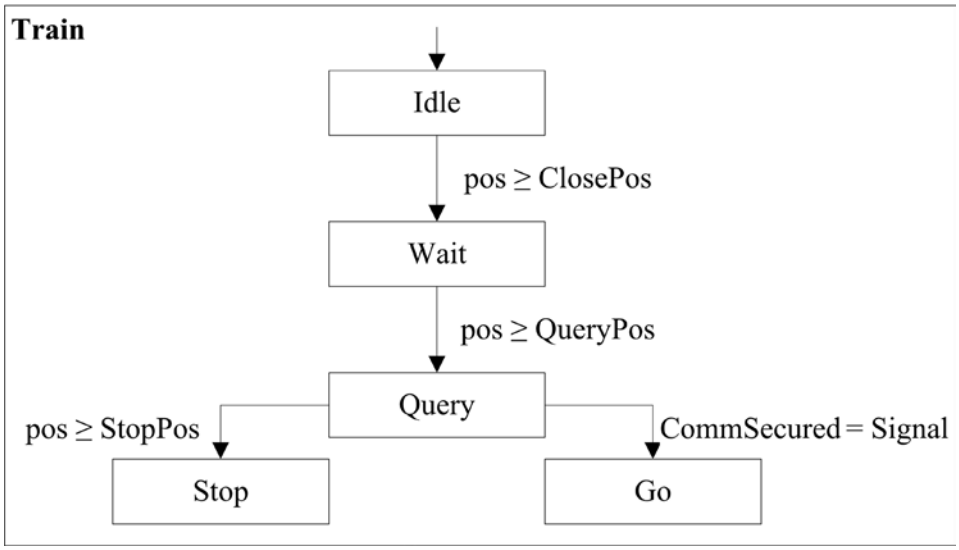


Table 1. The communication automata

Name	ϕ	Delay
CommClose	*Train = Idle ∧ pos ≥ ClosePos*	*CommDelay*
CommQuery	*Train = Wait ∧ pos ≥ QueryPos*	*CommDelay*
CommSecured	*Crossing = Closed ∧ CommQuery = Signal*	*CommDelay*
TimerOpen	*Crossing = Closed*	*CloseTimeout*
TimerClosing	*Crossing = Opening ∧ CommClose = Signal*	*ClosingDelay*

sage causing the barriers to begin closing, the message querying whether the crossing is closed, and the message informing the train that the crossing is secured. The first and the second ones are sent as soon as the train reaches the *Close* or the *Wait* positions, respectively. The third one is sent if the crossing is closed and the query message is received. The timer automata that delay the closing and opening of the barriers are also modeled by instantiating the program graph in Figure 10.

The program graph of the crossing, shown in Figure 11, leaves the *Opened* state when it receives the close message. It remains in the *Closing* state until the timeout has elapsed. It then switches into the *Closed* state where it remains until either the open timeout elapses or the sensor – which is connected by wire and therefore has no communication delay – detects the passage of the train; this detection is modeled by the train passing the sensor's position.

The system model presented here is functionally correct. Among others, the model satisfies the following properties that can be used as a sanity check to validate the model's quality and appropriateness:

- The train always moves with maximum speed.
- There is a system state in which the train has passed the crossing.

Figure 10. A generic program graph used for timeouts and communication delays

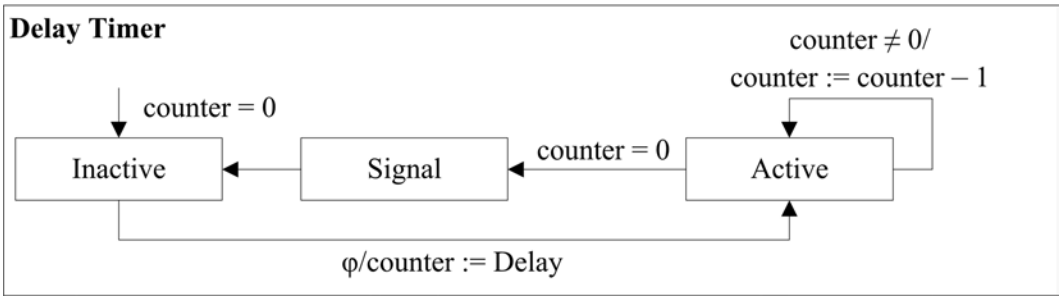
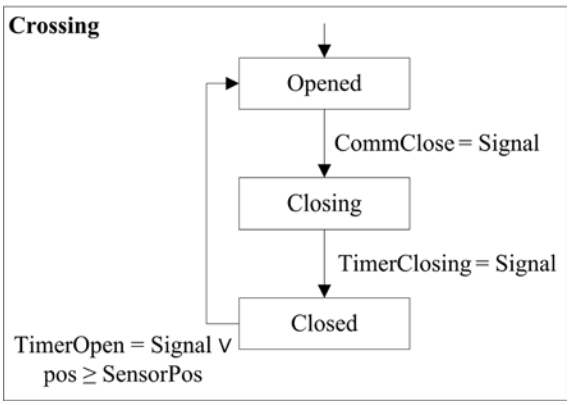


Figure 11. The program graph modeling the behavior of the crossing



- If the train is on the crossing, the crossing is closed.
- If the train has passed the sensor, the crossing is open.
- The crossing is closed as long as the train is between the crossing and the sensor.
- The transition relation of the system model is left-total.

All of these properties have been formalized (see also Section "Specification of System Properties") and have been successfully verified by NuSMV.

Formal Failure Models

So far, the system model only models the intended behavior of the system under consideration, whereas possible unwanted or erroneous behavior is ignored. For safety analysis purposes, however, it is imperative to model the undesired system behavior as well. Only then is it possible to formally examine which combinations of component failures lead to system hazards and to thoroughly analyze the respective cause-consequence relationships. While building a formal model that correctly describes the real system is already a difficult task, constructing such a model taking also all possible failure modes into account is even more challenging. Therefore, Subsection "Modeling Failures" advises a systematic approach where we incrementally extend the original failure-free model to include the relevant failure modes. Using the construction rules presented below, we can guarantee that the initial system behavior is preserved in the absence of failures; that is, the extended model is behaviorally equivalent to the original one as long as no failures occur. Subsection "Application to the Case Study" presents some possible failures that affect the safety of the case study and shows their integration into the formal model defined in the preceding section.

Modeling Failures

The main idea of the failure mode integration is to split the process into two parts: First, the occurrence pattern of a failure is modeled. Second, the actual local effects that the occurrence of a failure has on one of the system's components are added. The occurrence pattern describes under what circumstances a failure mode occurs. The most basic pattern is a transient one, where a failure mode can appear and disappear completely randomly. A typical transient failure is a malfunction of the radio communication in the railroad crossing case study, resulting in the loss of a communication message. Persistent failures, on the other hand, still allow the failure to occur randomly, but if and when it finally occurs, it stays forever. For instance, a broken motor that causes the barriers to be stuck somewhere in between the *Opened* and *Closed* states is a persistent failure, as it cannot repair itself. We use failure automata to model such occurrence patterns. A failure automaton has one distinguished state *No* with the semantics that the failure mode occurs if and only if the failure automaton is not in state *No*. Failure automata are allowed to be arbitrarily complex, making it possible to model more elaborate occurrence patterns as is shown below for the odometer failure. Figure 1 shows two program graphs modeling transient and persistent failures.

The direct effects of a failure mode are modeled by extending the original system model with the erroneous behavior. The concrete integration of a failure is application specific, but it must be ensured that the resulting model is a conservative extension of the original one. This guarantees that the integration of a failure does not affect the original behavior of the system as long as the failure does not occur. More formally, the original model and the extended model must be trace equivalent in the absence of failures. This section formally defines the notion of conservative extension, but also gives practical rules for integrating failure modes conservatively.

Definition 7: (Conservative Extension) Let $P : (V, S, \rightarrow, s_0, \phi_0)$ be a program graph with corresponding Kripke structure $K_P = (AP, S, R, L, I)$. A program graph $P' : (V', S', \rightarrow', s_0', \phi_0')$ is a conservative extension of P if $V \subseteq V'$, $S \subseteq S'$, $traces(P) \subseteq \{\Phi_0 \bigcap AP, \Phi_1 \bigcap AP, ... \mid \Phi_0, \Phi_1, ... \in traces(P')\}$.

In the above definition, the intersection $\Phi \bigcap AP$ restricts the propositions in Φ to all those which are expressible over the states and variables of the original program graph P.

The following theorem presents the rules for integrating a failure mode conservatively; if they are followed, the intended behavior of the system is guaranteed to be preserved, therefore making it unnecessary to individually perform a preservation proof for each integrated failure mode.

Theorem 1: (Conservative Failure Integration) Let $Sys : (V, S, \rightarrow, s_0, \phi_0)$ be a system model, P be the failure automaton of a failure mode, and $Sys' : (V', S', \rightarrow', s_0', \phi_0')$ be the extended system model obtained by first integrating the failure mode into Sys according to the following rules and then by synchronously composing the resulting modified system model and the failure automaton P.

Rule 1: New states $s \notin S$ and variables $v \notin V$ may be added arbitrarily.

Rule 2: Outgoing transitions from newly added states may be added arbitrarily.

Rule 3: Outgoing transitions from already existing states may be added if their guards are of the form $\phi \wedge P \neq No$, where ϕ is any formula in $F_{V'}$. Furthermore, guards ϕ of already existing transitions may be strengthened to $\phi \wedge P = No$.

Rule 4: The initial condition ϕ_0 must be extended to nonambiguously determine the initial value of each new variable. All actions α of all transitions may be extended to determine the updated values of the newly introduced variables.

Then Sys' is a conservative extension of Sys. Moreover, in the absence of the failure, that is, as long as $P = No$ holds, both models are trace equivalent:

$$traces(P) = \{\Phi_0 \bigcap AP, \Phi_1 \bigcap AP, ... \mid \Phi_0, \Phi_1, \\ ... \in traces(P') \wedge \forall i.(P = No) \in \Phi_i\}.$$

The proof of Theorem 1 is rather straightforward: States, transitions, and variables might be added, but all traces previously possible remain possible. The extended system model is constructed in such a way that new transitions can only be taken if the failure occurs. Thus, the system can never take any of these transitions and consequently also never enters any of the newly added states, because the failure automaton always remains in state *No* in all traces. Moreover, the initial state is not changed and the guards of all original transitions remain unaffected as long as the failure does not occur. As the changes to the initial condition and the actions only affect newly introduced variables, they also do not inhibit any of the original system behavior.

Rule 3 of Theorem 1 allows strengthening guards ϕ of already existing transitions to $\phi \wedge P = No$. It is often desirable to do so, either because the effect of a failure can be modeled by simply not allowing a transition to take place anymore if the failure occurs, or to enforce the effect of the failure. Suppose that a new transition is added to an already existing state that models the erroneous behavior of the system when the failure occurs. If the already existing transitions originating at that state are left unchanged, the system is still able to exhibit its intended behavior even in the presence of the failure. This is often not intended, as the occurrence of a failure should often deterministically have some undesired effect on the component.

Application to the Case Study

In order to model the erroneous behavior of the case study, we first have to identify the possible failure modes. Failure modes are typically provided by the component manufacturers. Alternatively, a safety engineer uses his intuition and experience in an informal process like FTA or FMEA to find all possible failures. Finding all relevant failure modes is imperative; the results of the analysis techniques presented in this chapter obviously cannot take failure modes into account that are not present in the extended system model. Neither is it a good idea to model all conceivable failures, as model checking and tool-assisted proofs become more time-consuming when more failures are taken into account. For example, it is physically possible that the barriers close without being asked to do so; however, this failure is irrelevant for the hazard we consider, so it can be omitted. Then again, this failure mode should not be omitted if we also study the hazard that the barriers should only be closed as long as necessary.

In this chapter, we consider the failure modes listed in Table 2. The occurrence behaviors of these failure modes are modeled by the transient or persistent failure program graphs shown in Figure 12. The only exception is the failure of the odometer: While it is a transient failure too, the program graph in Figure 13 additionally sets the delta between the actual speed and the value returned by the odometer to a random value in the range [-5..5], resulting in a speed delta of ± 27 km/h.

This list of failure modes is by no means exhaustive. As already mentioned above, we could allow the train to arbitrarily accelerate, thereby introducing an additional single point

Table 2. List of failure modes

Failure	Transient	Description
FailureBrakes	No	Malfunction of the brakes.
FailureOdometer	Yes	The discrepancies in the data returned by the odometer are greater than what the safety margin can compensate for.
FailureSecured	Yes	The crossing reports that it is secured even though the barriers are not closed.
FailureClose	Yes	The crossing does not close when asked to.
FailureOpen	Yes	The barriers open even though they are not instructed to do so, i.e., they should remain closed.
FailureStuck	No	The barriers are stuck somewhere in between the *Opened* and *Closed* state.
FailureComm	Yes	A malfunction of the radio communication results in the loss of communication messages.

Figure 12. Failure automata for transient and persistent failures

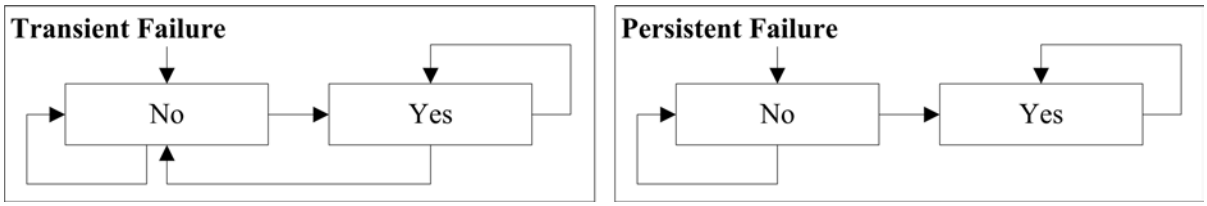
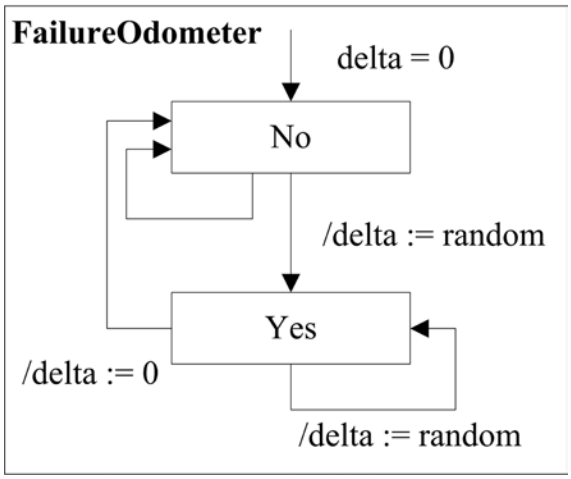


Figure 13. The program graph modeling the transient occurrence of the odometer failure, also determining the delta between the readings and the actual value



of failure. On the other hand, the timer TimerClosing is only added artificially to the system and has no real world counterpart. Therefore, it does not make sense to model failures affecting this timer. Moreover, we do not consider a failure of timer TimerOpen, because the failure mode "timer times out too soon" is subsumed by the failure FailureOpen; the failure mode "timer never times out" is not relevant for the considered hazard. For other systems in which timers are indeed physical components that are also safety-relevant, timer failures must be modeled to obtain meaningful results from the safety analysis techniques.

The seven failure modes are integrated into the functional model of the system as follows: *FailureStuck*, *FailureOpen*, and *FailureBrakes* are added conservatively to the original program graphs as shown in Figure 14. *FailureOdometer* affects only the calculations of the *Close*, *Query*, and *Stop* positions as these now also take the delta of the odometer's readings into account. This is also a conservative extension of the system model: For each transition referring to one of the three positions, an additional transition is added with

Figure 14. The extended program graphs of the brakes and the crossing

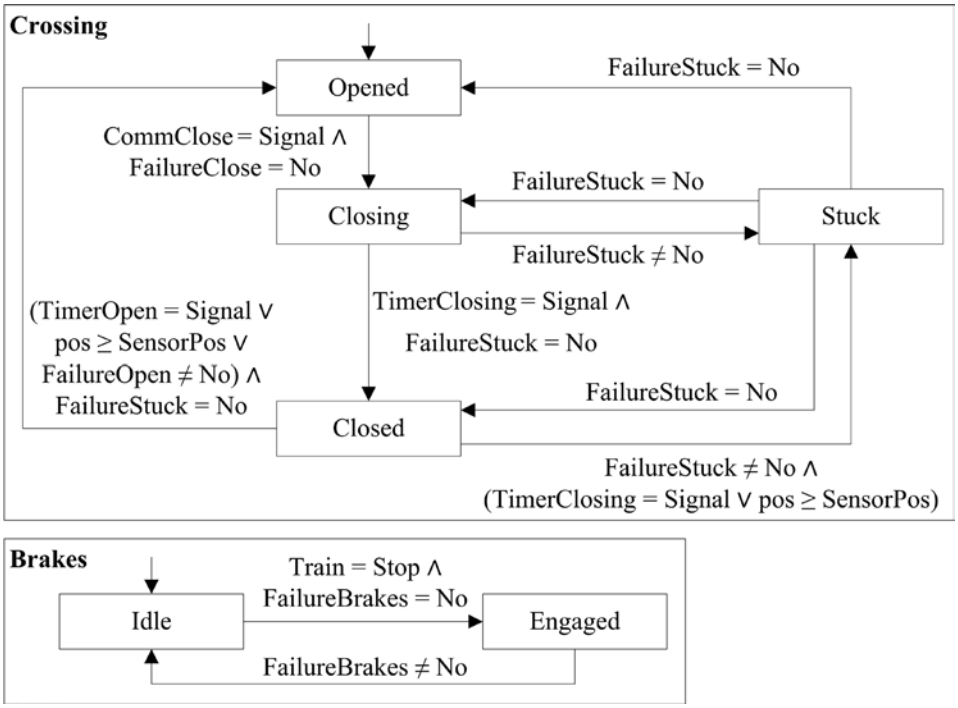


Table 3. Remaining failure modes

Name	ϕ
CommClose	*Train = Idle ∧ pos ≥ ClosePos ∧ FailureComm = No*
CommQuery	*Train = Wait ∧ pos ≥ QueryPos ∧ FailureComm = No*
CommSecured	*CommQuery = Signal ∧ FailureComm = No ∧ [Crossing = Closed ∨ (Crossing ≠ Closed ∧ FailureSecured ≠ No)]*

the same guard, replacing the original position with the adjusted one. The original transition can only be taken if the odometer failure does not occur, the new one can only be taken when the readings are off.

The remaining failure modes – *FailureSecured*, *FailureOpen*, and *FailureComm* – require changes to the activation conditions ϕ of the communication automata as summarized by Table 3.

The changes made to the activation conditions all follow the rules of conservative failure integration; the new activation condition of CommSecured actually combines two transitions from the program graph's *Inactive* state to its *Active* state; one for a successful sending of the message in the case that *FailureComm* does not occur, and one for the case that the crossing erroneously sends the message. The activation conditions of both transitions are combined disjunctively into a single transition, and the resulting activation condition is transformed into the formula shown above.

In Section "Formal System Model" advises to perform some sanity checks to verify that the model indeed exhibits the intended behavior of the real system. This can also be done after the integration of the failure modes to ensure that no mistakes were made and that the constants still have meaningful values. For instance, the odom-

eter failure should have an effect on the system, but if the safety margin is set to too high a value, this might not be the case. Among others, the extended system model presented in this section satisfies the following properties:

- There is a system state in which the train has stopped before reaching the crossing.
- Conversely, there is a system state in which the train has passed the crossing.
- The barriers can indeed get stuck.
- The intended behavior is preserved, i.e., it is still possible that the train is on the crossing and the crossing is closed.
- The transition relation of the extended system model is left-total.

Again, these properties can be formalized as temporal logic formulae (see next section) and verified using the NuSMV model checker.

SPECIFICATION OF SYSTEM PROPERTIES

The two preceding sections list several properties for the purpose of checking whether the behavior of the model of the case study matches the intended behavior described by the informal specification of the system. These properties are given in plain English and therefore do not allow formal reasoning about their validity. There is a wide variety of specification languages for formalizing system properties (see Section "A Brief Introduction to Formal Methods"), with temporal logics being one possible choice. We introduce the temporal logic CTL* in the following to formally specify properties over the model of the case study; more precisely, over the Kripke structure induced by the model. An appropriate tool like NuSMV or KIV can then be used to verify that the model indeed satisfies these properties. Note that we do not distinguish between the formal system model and the extended system model for the remainder of this section; in fact, the following discussion is of rather abstract nature and applies to any kind of system model whose semantics can be given as a Kripke structure.

Temporal logics allow expressing events like failures, hazards, and intermediate events of a fault tree (see next section) over the system model. In contrast to what the term "temporal" might suggest, there is no relationship between the real-time behavior of a system and CTL*; rather, CTL* is based on an abstract time model that specifies the relative order of events. For instance, properties like "a collision cannot occur as long as only the brakes fail" or "a message is sent and after a finite amount of time it is received" can be expressed. Other kinds of temporal logics exist which are also able to express properties that refer to the real-world timing or duration of events like "it takes one minute to secure the crossing". In CTL*, however, an event can only indirectly imply its duration by stating the number of consecutive states for which the property must hold. As mentioned above, each system step is assumed to take some time, e.g., each system step of the case study is assumed to take two seconds, so if a property holds for three consecutive system steps, it actually holds for six seconds. Furthermore, it is important to distinguish between temporal and state events. The former involve some notion of time, whereas the latter do not and can therefore be expressed in plain propositional logic. In other words, temporal events talk about different states of the system at different points in time, as opposed to state events which are only concerned about the current state of the system; such a distinction is typical for modal logics like CTL*. In the following, we are mostly concerned with state events, but we also discuss some issues with temporal events and how to deal with them because they are sometimes encountered when assessing system safety. For instance, the pressure tank example in the FTA handbook (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002) analyzes the hazard of a tank rupture if the tank is filled continuously for more

than sixty seconds. "Filling for more than sixty seconds" clearly is a temporal event.

Many model checkers (including NuSMV) only support Computation Tree Logic (CTL) or Linear Temporal Logic (LTL) instead of CTL*. CTL* subsumes both CTL and LTL, but model checkers generally prefer CTL over both LTL and CTL* as CTL model checking algorithms are more efficient. However, the expressiveness of CTL and LTL is incomparable, so sometimes it is unavoidable to use LTL to express certain properties. The techniques presented in the subsequent sections rely on CTL exclusively; CTL* is only needed to state theorems 2 and 3. It is often difficult to adequately express an informal statement as a CTL formula. However, many properties encountered in the context of safety analysis can be derived systematically: For instance, structured approaches for the systematic formalization of informal specifications with temporal logics are available in the FSAP/NuSMV-SA tool (Bozzano & Villafiorita, 2007) or via the "safety pattern" described in (Bitsch, 2001).

The following subsection gives an overview of CTL*. Subsection "Causes and Consequences" discusses the characteristics of cause-consequence relationships which underlie our analysis techniques. Finally, Subsection "Application to the Case Study" discusses the formalization of some of the properties mentioned in the preceding sections.

Definition of the Temporal Logic CTL*

We give the syntax and semantics of CTL* in a somewhat informal way and assume that the reader is familiar with the formal semantics of propositional logic connectives such as conjunction $\wedge$ and negation $\neg$; a more formal introduction can be found in (Baier & Katoen, 2008). A CTL* formula φ is evaluated over a path $\sigma = s_0, s_1, \ldots$ of a Kripke structure $K = (AP, S, R, L, I)$. We write $\sigma \models \varphi$ to mean that formula φ is valid in path σ. In particular, a proposition $p \in AP$ is valid if it is valid in the first state s_0 of the path, that is, if it is contained in the state's labeling $L(s_0)$. For a given Kripke structure K, we write $K \models \varphi$ to mean that the formula φ is valid for K, i.e., $\sigma \models \varphi$ holds for all paths $\sigma \in paths(K)$. In contrast to propositional logic, CTL* is a modal logic that can refer to different system states. A propositional logic formula is always evaluated in the current system state, whereas a CTL* formula containing modal (temporal) operators is evaluated over a sequence of states, as it typically refers to properties that will hold in the future. Hence, propositional logic suffices to formalize state events, whereas temporal events require some temporal logic such as CTL*.

Definition 8: (CTL*) Let $\varphi, \varphi_1, \varphi_2$ be CTL* formulae and $\sigma = s_0, s_1, \ldots$ be a path of a Kripke structure K. Let $\sigma_{[i\ldots]} = s_i, \ldots$ denote the postfix of the path σ starting at state s_i. The operators defined by CTL* and their semantics are as follows:

- *"Next"* – In the following state
  $\sigma \models X\varphi \quad :\Leftrightarrow \quad \sigma_{[1\ldots]} \models \varphi$
- *"Finally"* – In some future state
  $\sigma \models F\varphi \quad :\Leftrightarrow$
  for some $i \geq 0$, $\sigma_{[i\ldots]} \models \varphi$
- *"Globally"* – From now on forever
  $\sigma \models G\varphi \quad :\Leftrightarrow$
  for all $i \geq 0$, $\sigma_{[i\ldots]} \models \varphi$
- "Until" – φ_1 holds up to the point where φ_2 holds; φ_2 must hold in some future state
  $\sigma \models \varphi_1 \; U \; \varphi_2 \quad :\Leftrightarrow$
  for some $i \geq 0$, $\sigma_{[i\ldots]} \models \varphi_2$
  and for all $0 \leq j < i$, $\sigma_{[j\ldots]} \models \varphi_1$
- "Precedes" – φ_2 must not occur without φ_1 occurring before

$\varphi_1 \; P \; \varphi_2 \quad :\Leftrightarrow$
$\neg((\neg\varphi_1) \; U \; (\varphi_2 \wedge \neg\varphi_1))$

- "For all paths" – Formula φ holds in all possible paths originating at the current system state
  $\sigma \models A\varphi \quad :\Leftrightarrow$
  for all paths
  $\sigma' = s_0, s'_1, \ldots \in paths(K), \sigma' \models \varphi$
- "There is a path" – There is at least one possible path originating at the current system state in which formula φ holds $E\varphi \quad :\Leftrightarrow \quad \neg A \neg \varphi$

In contrast to CTL*, CTL requires that path quantifiers (E, A) and temporal operators (X, F, G, U, P) always come in pairs; for example, AF, EG, or EU. Note that the expansion of $A(\varphi_1 P \varphi_2)$ to $A\neg((\neg\varphi_1) \; U \; (\varphi_2 \wedge \neg\varphi_1))$ is equivalent to $\neg E((\neg\varphi_1) \; U \; (\varphi_2 \wedge \neg\varphi_1))$ and is thus a valid CTL formula. The same argument can be made for $E(\varphi_1 P \varphi_2)$. By contrast, LTL disallows the use of path quantifiers; all LTL formulae are implicitly quantified over all paths.

Consider the following two abstract examples of system properties formalized as CTL formulae: $AG \; EF \; FailSafe$ is a formalization of "The system can always (AG) reach (EF) a fail-safe state", where the proposition $FailSafe$ is true if and only if the system is in the fail-safe state. $\neg E(NoFailures \; U \; Hazard)$ has the informal meaning of "There is no path ($\neg E$) in which eventually a hazard occurs but no failures occurred before (U)". Subsection "Application to the Case Study" illustrates how some of the aforementioned properties of the case study can be formally expressed in CTL.

Causes and Consequences

In the context of safety analysis, causes are (combinations of) component failures. Depending on the point of view, intermediate events of a fault tree (see next section) are either causes or consequences. The ultimate consequence is the hazard under consideration. This subsection discusses the notions of potential and effective causes for some consequence, which formally characterize the cause-consequence relationships underlying both of our analysis techniques.

An event is a potential cause of another event, the consequence, if the former occurs completely before the latter. For instance, if the event "barriers open unexpectedly" occurs shortly before the train reaches the crossing, it is a potential cause for the hazard (in fact, it also is an effective cause, see below). On the other hand, if the brakes of the train are destroyed in the course of a collision, the event "failure of the brakes" certainly was not a potential cause, but a consequence of the crash. Such a relationship between two events is formalized by the formula $\varphi \, P \, \psi$, where φ denotes the potential cause and ψ describes the consequence. For a state event φ, this formula guarantees that φ occurs completely before or at the same time as ψ. However, for the general case where φ is either a state event or a temporal event, the formula is inadequate: For a temporal event φ, the formula $\varphi \, P \, \psi$ only guarantees the beginning of φ but does not necessarily guarantee its completion before ψ occurs.

To illustrate this point, consider the pressure tank example in (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002): A rupture of the tank ψ is potentially caused by continuous pumping for more than sixty seconds. This potential cause can be expressed by the CTL* formula $\varphi = p \wedge Xp \wedge XXp \wedge \ldots \wedge X^{60}p$, with p being some proposition that holds if and only if the pump is running and with one system step corresponding to one second. Clearly, φ is a temporal event with the informal meaning that p must hold for sixty consecutive states. The formula $\varphi \, P \, \psi$ might spuriously identify φ as a potential cause for ψ: Suppose a rupture ψ occurs at state s_i of some path σ. If p holds in states s_i, s_{i+1},

…, and s_{i+60}, φ holds in s_i and consequently so does $\varphi P \psi$. However, proposition p is required to hold for sixty seconds (sixty consecutive states) in order to cause the hazard, so it cannot have caused ψ if it only held for one second (one state). If there is no other path that adequately respects the intended relationship between the two events, φ would unjustifiably be characterized as a potential cause of ψ. Consequently, $\varphi P \psi$ is inadequate when temporal causes are involved and thus all causes and consequences are required to be propositional formulae that only refer to the current system state, and not to future or past ones. Early work on formal specifications of cause-consequence relationships in the context of fault tree analysis such as (Bruns & Anderson, 1993) or (Hansen, Ravn, & Stavridou, 1994) has typically ignored the issue with temporal events or considered Boolean semantics only (Hansen, Ravn, & Stavridou, 1998).

As temporal events are sometimes encountered in safety analysis, there must be a way to cope with them. It is indeed possible to avoid the aforementioned issue by introducing observer automata into the system model; one for each temporal event. Such an observer automaton is constructed so that a distinguished acceptance state is entered once the temporal event φ holds completely for the first time. Let the propositional formula φ_{acc} denote the (state) event that is satisfied if the observer automaton is in its acceptance state. $\varphi_{acc} P \psi$ then adequately decides whether the temporal event φ is a potential cause for consequence ψ. Referring back to the pressure tank example above where the temporal event $\varphi = p \wedge Xp \wedge XXp \wedge ... \wedge X^{60}p$ was in fact not a potential cause for ψ but identified as such, the corresponding observer automaton enters its acceptance state once it has observed that proposition p held for sixty seconds (sixty consecutive states). $\varphi_{acc} P \psi$ is then only satisfied if p held for at least sixty system steps before ψ is satisfied. Obviously, the aforementioned path σ does not fulfill $\varphi_{acc} P \psi$ and as by assumption all other paths satisfy neither $\varphi P \psi$ nor $\varphi_{acc} P \psi$, $\varphi_{acc} P \psi$ does not spuriously identify φ as a potential cause for ψ. Observer automata thus provide a general mechanism to reduce temporal events to state events. In the following, causes and consequences are always expressed as propositional logic formulae, that is, as state events. We assume that temporal events are always reduced to state events by the introduction of an observer automaton.

An event is an effective cause for some consequence if it is a potential cause for the consequence and if it there is at least one possible scenario in which the event really causes the consequence. This emphasizes the fact that none or not all occurrences of a cause definitely result in the consequence; there might be situations in which the consequence is avoided because of sheer luck or for some other (systematical) reason resulting from the complex interactions of the system's components. Both of our analysis techniques yield potential causes for the hazard under consideration, but only Deductive Cause Consequence Analysis guarantees that all identified causes are effective, therefore yielding more optimal results in contrast to Formal Fault Tree Analysis. This difference is described in further detail in the following section.

Application to the Case Study

The model of the case study without the undesired behavior satisfies the property "The train always moves with maximum speed." as mentioned in Section "Formal System Model". In CTL, this property is expressed as

$$AG\ speed = MaxSpeed \tag{1}$$

The more complex property "If the train has passed the sensor, the crossing is open." can be formalized as

$$AG\ (pos \geq SensorPos \rightarrow AX\ Crossing = Opened) \quad (2)$$

The second temporal operator AX is necessary because the crossing does not immediately open after the train has passed the sensor. The reaction of a program graph to a state change of another program is always delayed by one time step, hence the crossing is still closed when the sensor detects the train. Therefore, property (2) does not hold if the AX operator is omitted.

The property "There is a system state in which the train has stopped before reaching the crossing." cannot be shown for the formal system model, as the train has no reason to ever stop in the absence of failures. For the extended system model, however, the property does hold. It is formalized as

$$EF\ (speed = 0 \wedge pos < CrossingPos)$$

By contrast, it is not possible to verify property (1) for the extended system model, as now there are paths that cause the train to stop. Property (2) also does not hold in the presence of failures, only a weakened version of the property does:

$$EG\ (pos \geq SensorPos \rightarrow AX\ Crossing = Opened)$$

The first path quantifier is replaced by existential quantification as now there are states in which the crossing is not in the *Opened* state after the train has reached the sensor; e.g., the barriers might be stuck, preventing them from closing.

ANALYSIS METHODS

In this section, we present two formal safety analysis techniques: Formal Fault Tree Analysis (FFTA) and Deductive Cause Consequence Analysis (DCCA). We first introduce FFTA and show a formal fault tree of the case study. We then continue to explain the theory behind DCCA and apply the technique to the case study. Additionally, we show how DCCA relates to functional correctness, FMEA, and FFTA.

Formal Fault Tree Analysis

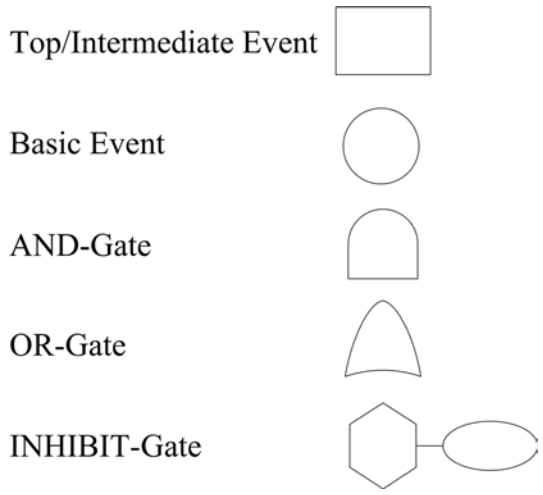
Fault Tree Analysis (FTA) is a widely used safety analysis technique (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002). We assume the reader to be familiar with traditional fault tree analysis and give only a brief overview of the technique in the next subsection. Subsection "Formalization of Fault Trees" then discusses how an informal fault tree is formalized, and Subsection "Application to the Case Study" presents a formal fault tree of the case study.

Brief Overview of Traditional Fault Tree Analysis

FTA systematically supports safety engineers in assessing system safety. A fault tree specifies an undesired state of a system, usually a hazard, and then systematically lists all causes that may lead to this state. The analysis is recursive, that is, all causes of the hazard are again examined to find their causes. The analysis is completed once all causes are fully identified, i.e., all basic component failures possibly leading to the hazard are found, resulting in a tree structure where the hazard is the root and the component failures are the leaves.

Hazards and causes are typically called events in the context of fault tree analysis. The hazard is the top event of the fault tree, component failures are basic events, and all events in between those are called intermediate events. In traditional FTA, events are described using natural language; FFTA assigns a formula over the extended system model to each event as described in the next subsection. Events should always describe a fault, that is, an undesired system state as indicated by "Ground Rule 1" given in (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002, p. 50).

Figure 15. The most common fault tree symbols



The causes of an event and the event itself are connected by gates and constitute a cause-consequence relationship as explained in the preceding section. The event describing the consequence of a gate is called the output event, whereas the causes are called input events. The type of the gate influences the relationship of input and output events: AND-gates require all input events to occur before the output event occurs, OR-gates require at least one of the input events to occur, and in the case of an INHIBIT-gate, the output event only occurs if the input event occurs and the enabling condition, noted next to the gate, holds. AND-gates are similar to INHIBIT-gates, the only difference being that the enabling conditions of INHIBIT-gates usually do not describe an undesired system state. Figure 15 shows the graphical notation of the events and the gates. Several other gate types are defined in (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002), but we restrict this discussion to the most common ones.

The basic events in the leaves of a fault tree represent the failure modes of the system, a combination of which might trigger the top event. A combination of basic events that might lead to the hazard under consideration is called a cut set. We are mostly interested in minimal cut sets, as they guarantee that they cannot cause the top event as long as at least one of their events does not occur. Hence, minimal cut sets help to identify failure modes that, if prevented, make the system safer: Minimal cut sets with only one element indicate a single point of failure and are thus candidates for further safeguarding. On the other hand, a basic event that is an element of all minimal cut sets should also be taken further care of, as preventing this failure or at least making it less likely to happen significantly decreases the probability of the top event. Cut sets can be deduced from the structure of the fault tree by interpreting AND-gates and OR-gates as conjunctions and disjunctions, the basic events as atomic formulae, and the fault tree as the syntax tree of a Boolean formula. The cut sets are obtained from this formula by transforming it into its disjunctive normal form (DNF). All resulting cut sets can then be checked for minimality. (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002, pp. 157-161) provides more details.

Formalization of Fault Trees

This subsection presents a formalization of Fault Tree Analysis along with the verification conditions that guarantee the completeness of the fault tree relative to a given model. Here, completeness means that all relevant failure modes for a hazard are considered in the fault tree. Obviously, if a relevant failure mode is omitted during the construction of the extended system model, the resulting fault tree can nonetheless be proven complete. As with all formal analysis techniques, the analysis is only as good as the model it is based on. Deductive Cause Consequence Analysis, presented below, suffers from the same issue. This highlights the importance of the correct construction of the functional system model and the correct identification and integration of all relevant failure modes.

To perform a FFTA, a safety engineer proceeds as follows: The informal system specification

is first formalized into a model of the intended behavior of the system and then all relevant failures modes are identified and integrated into the system model, yielding the extended system model. Traditional safety analysis techniques such as HAZOP or FMEA help identifying relevant failure modes and hazards. An informal FTA is conducted for each relevant hazard which is subsequently formalized as described below. This formalization of the informal fault tree makes it possible to check its completeness, ensuring that the fault tree contains all effective causes of the analyzed hazard.

A formal fault tree assigns a CTL formula over the extended system model to each event, replacing its description in natural language. Additionally, all gates are given a precise semantics by CTL formulae that combine the input and output events. FFTA can also be performed using LTL, so it can be adjusted to the property specification language supported by the tool used for the verification. As we use NuSMV to check the completeness of the formal fault tree, we are free to use either of these but focus exclusively on CTL in this chapter; the corresponding LTL formulae can be obtained by simply removing the "for all paths" quantifier.

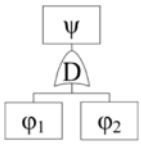
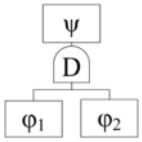
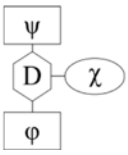
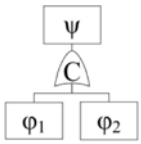
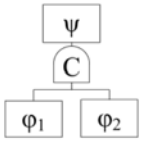
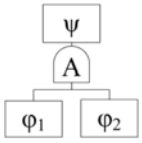
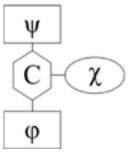
Previous attempts at giving a formal semantics to the gates of fault trees violated the aforementioned characteristics of the cause-consequence relationships. (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002) and (Leitsch, 1995) define the semantics of OR- and AND-gates simply as Boolean disjunctions or conjunctions of the input events, respectively. The Boolean semantics is only correct if the gate represents a simple decomposition of the output event into the input events. For instance, the proposed Boolean semantics is adequate for the output event "train on crossing, barriers not closed", which decomposes into two input events "train on crossing, barriers not closed, no secured message received" and "train on crossing, barriers not closed, secured message received" connected by an OR-gate. For cause-consequence relationships, on the other hand, the input events must happen before the output event to correctly model the relationship. This, however, is not guaranteed by the Boolean semantics. Consequently, more elaborate gate types than those listed in Figure 15 are required for formal FTA to correctly describe temporal events and the resulting cause-consequence relationships.

We therefore differentiate between decomposition gates which we give the Boolean semantics and cause-consequence gates that have a more complex semantics, taking timing considerations into account; this distinction is also proposed informally in (Górski, 1994). All in all, FFTA distinguishes between three types of decomposition gates (D-AND, D-OR, and D-INHIBIT) and four types of cause-consequence gates (C-AND, C-OR, and C-INHIBIT, the former of which is additionally subdivided into synchronous and asynchronous C-AND-gates). A synchronous C-AND-gate requires all input events to occur at the same time before the consequence, whereas asynchronous C-AND-gates only require that all input events occur before the consequence, but they do not necessarily have to occur simultaneously. In our graphical notation of the gates, decomposition and cause-consequence gates are distinguished by marking the gate symbol with either "D" or "C" – in the case of C-AND-gates, "C" designates a synchronous gate whereas "A" denotes an asynchronous one.

The Table 4 lists the CTL formulae that define the semantics of each of the seven gate types of FFTA. The intuition behind these formulae is described.

The traditional Boolean semantics of decomposition gates is expressed by an implication: The occurrence of the consequence ψ implies the occurrence of the conjunction or disjunction of the input events ϕ_1 and ϕ_2 ; that is, whenever the consequence occurs, at least one of the causes (D-OR-gate) or both causes (D-AND-gate) occur as well; otherwise there would be further causes and the consequence would not be adequately

Table 4. CTL formulae

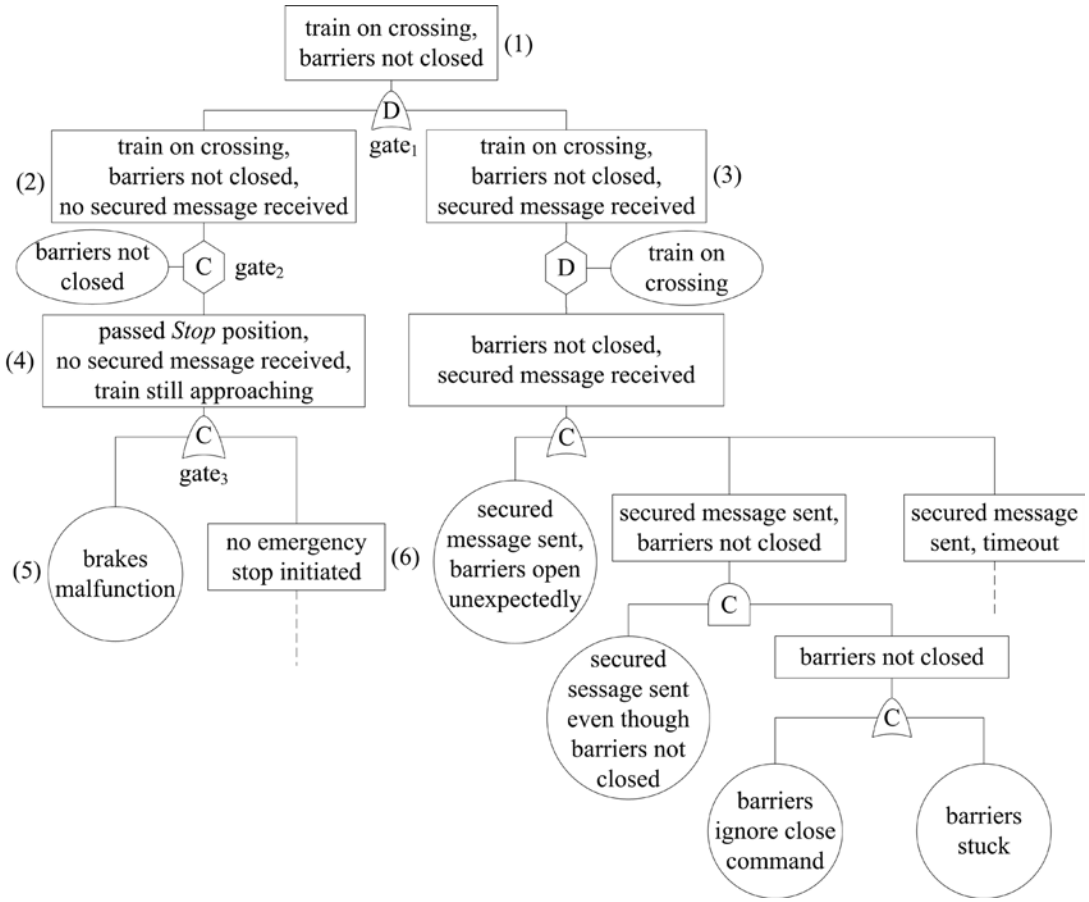
Gate	CTL formula
D-OR 	$AG(\psi \rightarrow \phi_1 \vee \phi_2)$
D-AND 	$AG(\psi \rightarrow \phi_1 \wedge \phi_2)$
D-INHIBIT 	$AG(\psi \rightarrow \phi \wedge \chi)$
C-OR 	$A((\phi_1 \vee \phi_2)P\psi)$
Synchronous C-AND 	$A((\phi_1 \wedge \phi_2)P\psi)$
Asynchronous C-AND 	$A(\phi_1 P\psi) \wedge A(\phi_2 P\psi)$
C-INHIBIT 	$A(\phi\, P\psi) \wedge \ A(\chi\, P\psi)$

described. The formula of D-INHIBIT-gates is equivalent to the one of D-AND-gates, as the difference between the second cause and the enabling condition is just of methodological nature – the second input event describes a fault, whereas the enabling condition does not – and has no formal implications. As the underlying system is dynamic, the implication must hold in all states of all paths of the system.

Cause-consequence gates take timing considerations into account; this is formally expressed by the *precedes* operator. The input events must occur before or at the same time the consequence occurs on all paths of the system. For C-OR-gates and synchronous C-AND-gates, either one of the input events or both input events must occur in some system state before or simultaneously with the output event. In particular, $\phi_1 \wedge \phi_2$ only holds if both input events occur in the same system state. The formula of asynchronous C-AND-gates, on the other hand, requires only that both input events precede the occurrence of the output event, but they are allowed to occur at any time before the output event and not necessarily simultaneously. The semantics of C-INHIBIT-gates is analogous to the semantics of asynchronous C-AND-gates, again with the difference being that the enabling condition does not describe a fault. It is possible to distinguish between synchronous and asynchronous C-INHIBIT-gates similar to the distinction between synchronous and asynchronous C-AND-gates, but we do not do so as the enabling condition and the input event usually do not occur simultaneously.

A formal fault tree is complete if all proof obligations resulting from the gates that are used in the construction of the fault tree can be verified. Completeness guarantees that all relevant causes of a hazard are considered, that is, the hazard cannot occur without some cause mentioned in the fault tree also occurring. Thus, it is sufficient to prevent one basic event of each minimal cut set to avoid the hazard altogether. The following completeness theorem formally justifies the use of minimal cut sets in safety analysis, in particular for cases where timing conditions are relevant. The cut sets of a formal fault tree are obtained by first replacing all cause-consequence gates with decomposition gates and then by using the same algorithm as for informal fault trees, that

Figure 16. Partial informal fault tree for a hazard of the radio-based railroad crossing case study



is, by constructing the DNF of the fault tree with traditional Boolean semantics.

Theorem 2: (Completeness of FFTA) Let K be a Kripke structure describing the extended system model and T be a formal fault tree with top event H . T is complete if the following CTL* formula holds:

$$K \models A\left(\left(\bigwedge_{\Gamma \in mcut(T)} \neg \bigwedge_{\delta \in \Gamma} F\delta\right) \rightarrow G\ \neg H\right)$$

where $mcut(T)$ denotes the set of all minimal cut sets Γ of the given fault tree T.

The theorem is proven using structural induction over the size of the fault tree. The basic fact underlying the proof is the transitivity of the cause-consequence relation in both directions, which is proven using the semantics of the involved basic temporal operators.

Application to the Case Study

Figure 16 shows a partial informal fault tree for the case study. The fault tree already distinguishes between decomposition and cause-consequence gates, but the events are still described in plain English.

The top event (1), the hazard of a collision, is defined as the situation that a train is on the crossing whilst the barriers are not closed:

$$e1 = pos \leq CrossingPos \wedge CrossingPos < pos + speed \wedge Crossing \neq Closed.$$

Because of the discretization of the position, we cannot simply write $pos = CrossingPos$. The formalization above ensures that the event occurs whenever the train is in front of or on the crossing in the current step and behind the crossing in the next step. We identify two possible causes for this hazard, one of which is enough to trigger the top event. Consequently, the two input events are connected by an OR-gate. Event (2) corresponds to a malfunction of the train, as it passes the non-closed barriers even though it did not receive a secured message. Event (3) describes a failure of the crossing, as it is not closed even though it sent the secured message – in our model, a message can only be received if it has in fact been sent earlier. Since there are no cause-consequence relationships between the input and output events of $gate_1$, we use a decomposition gate in this case.

The formalizations of events (2) and (3) cannot directly express whether the train has received the secured message. In the model, the reception of the message is represented by the CommSecured automaton being in the *Signal* state – where it remains only for one step of the system. Hence, in the situation described by the top event, the reception of the message lies in the past and cannot be expressed by any CTL formula. We therefore add an observer automaton as introduced in Section "Specification of System Properties" called SecuredReceivedObserver to the system that enters the *Yes* state as soon as the secured message is received. Thus, the formalization of events (2) and (3) are:

$$e2 = e1 \wedge SecuredReceivedObserver = No$$
$$e3 = e1 \wedge SecuredReceivedObserver = Yes$$

Event (4) is connected to event (2) via an INHIBIT-gate. As time elapses between events (2) and (4), a cause-consequence INHIBIT-gate is used. The set of states considered faulty by event (4) is larger than the one of event (2). Here, we drop the condition that the barriers are not closed, therefore enlarging the set of all states considered faulty by event (4). That is, we consider event (4) to be a failure even though this situation is not a problem on its own; the system's safety is only compromised if the barriers are not closed in addition to the conditions of event (4). The enabling condition explicitly specifies which additional states are considered faulty and can be formally expressed as $Crossing \neq Closed$; note that it is not a fault on its own. Event (4) is formally described by

$$e4 = pos \geq StopPos \wedge SecuredReceivedObserver = No \wedge speed > 0$$

$Gate_3$ is another cause-consequence OR-gate, as both input events (5) and (6) happen before or at the same time as event (4). The malfunction of the brakes corresponds to failure mode *FailureBrakes* and is thus a basic event formalized as

$$e5 = FailureBrakes = Yes$$

Event (6) is described by the following formula, but like event (3) it is not further analyzed in this chapter:

$$e6 = Train \neq Stop$$

The proof obligations of the gates shown in the fault tree can be verified using model checking. Therefore, the partial fault tree shown in Figure 16 is complete, that is, a collision does not occur if none of the basic or intermediate events occur. It is interesting to note that a malfunction of the brakes is a single point of failure according to this fault tree. However, it is in fact not a problem if the brakes fail as long as everything else works as intended: Whilst no other failures occur, the crossing will be closed once the train passes it. It is of course possible to construct the fault tree in such a way that the brakes are no longer identified as a single point of failure. This is difficult, however,

as a fault tree gives no indication as to whether all identified cut sets are indeed effective causes.

DEDUCTIVE CAUSE CONSEQUENCE ANALYSIS

This section describes another formal analysis method for determining relationships between causes and consequences. Deductive Cause Consequence Analysis (DCCA) is a formal analysis method which allows fully automated computation of relationships between component failures and system hazards. As for FFTA, an extended system model with integrated failure modes is required over which the hazard under consideration H is formalized using propositional logic connectives only. For better readability, the remainder of this section uses the name of a failure automaton P to describe the formula $P \neq No$, i.e., to express that the failure has occurred. Δ denotes the set of all such formulae of all failure modes. We define the notion of criticality of a combination of failure modes that may lead to the hazard as follows:

Definition 8: (Critical Set/Minimal Critical Set) Let K be a Kripke structure, H be a formula over K describing a hazard, and Δ be a set of failure modes. A subset of component failures $\Gamma \subseteq \Delta$ is called critical for H if

$$K \models E(\lambda \; U \; H) \qquad \text{where } \lambda := \bigwedge_{\delta \in (\Delta \setminus \Gamma)} \neg \delta \; .$$

Γ is a minimal critical set if it is critical but no proper subset of Γ is critical.

The criticality of a set of failure modes translates into natural language as follows: There is a path such that the hazard H occurs without the previous occurrence of any failures except those which are in the critical set. Note that criticality alone is not sufficient to ensure that all causes are relevant. It is possible that a critical set comprises failure modes which have no influence on the hazard. Therefore, the notion of minimal critical sets also requires that no proper subset of a critical set is critical. Minimal critical sets are effective causes for the hazard. Note that $\lambda \; U \; (H \wedge \lambda)$ is be equivalent to $\neg(\lambda \; P \; H)$. DCCA requires the failures to strictly precede the consequence. If a hazard and a component failure occur exactly at the same time, then DCCA does not consider this failure as a potential cause of the hazard.

A DCCA for a hazard H is the (fully automated) computation of all minimal critical sets. From an algorithmic point of view, testing all possible combinations of failures by brute force would require an effort exponential in the number of failure modes. In practice, the number of proofs can usually be reduced significantly by exploiting the fact that the criticality property is monotone with respect to set inclusion, i.e., $\forall \Gamma_1, \Gamma_2 \subseteq \Delta . \Gamma_1 \subseteq \Gamma_2 \Rightarrow (\Gamma_1 \text{ is critical} \Rightarrow \Gamma_2 \text{ is critical})$. Additionally, traditional analysis techniques are often used to quickly obtain preliminary results prior to performing a more time-consuming DCCA. This can help to further reduce the effort of DCCA, because the informal techniques often yield good "initial guesses" for potential candidates of minimal critical sets.

A completeness theorem similar to that of formal FTA can be proven for minimal critical sets determined with DCCA.

Theorem 3: (Completeness of DCCA) Let K be a Kripke structure, H be a hazard, and Δ be a set of failure modes. The analysis is complete if the following CTL* formula holds:

$$K \models A\left(\left(\bigwedge_{\Gamma \in mcrit(K,\Delta,H)} \neg \bigwedge_{\delta \in \Gamma} F\delta\right) \rightarrow G \neg H\right)$$

where $mcrit(K, \Delta, H)$ denotes the set of all minimal critical sets Γ of the system K for a hazard H and a set of failure modes Δ. The set $mcrit(K, \Delta, H)$ is the result of a DCCA.

Table 5. Minimal critical sets

{FailureOpen}	*{FailureComm, FailureOdometer}*
{FailureClose, FailureBrakes}	*{FailureComm, FailureBrakes}*
{FailureStuck, FailureBrakes}	*{FailureClose, FailureOdometer}*
{FailureStuck, FailureSecured}	*{FailureStuck, FailureOdometer}*
{FailureClose, FailureSecured}	*{FailureComm, FailureSecured}*

Informally, the completeness theorem states that if at least one failure mode of each minimal critical set can be avoided, then the hazard will never occur. It follows that if the hazard has occurred, then at least one minimal critical set has occurred strictly before. The proof obligations of DCCA guarantee that for every minimal critical set, there is at least one path in which the hazard is caused only by the failure modes in this set. There are no minimal critical sets that cannot cause the hazard.

Application to the Case Study

DCCA identifies the following minimal critical sets for the case study. It is first verified that the hazard cannot occur in the absence of any failures, because in that case, no further minimal critical sets would be found due to the monotony of the criticality property. There is one single point of failure, and nine combinations of two failure modes that may lead to the hazard. Due to the monotony of the critical sets only two sets containing three failure modes have to be analyzed. These are not critical and no further analysis is required for combinations of more than three failures due to monotony.

The minimal critical sets of the case study for the failure modes described in Section “Formal Failure Models” are:

Comparison to other Safety Analysis Techniques

DCCA is a unifying theory for some traditional safety analysis techniques, depending on the number of elements in the analyzed set of failure modes. In the following, we discuss the relation to verification of functional correctness, Failure Modes and Effects Analysis and Formal Fault Tree Analysis.

Functional Correctness

If the empty set $\Gamma = \{\}$ of failure modes is examined, the proof obligation of minimal criticality corresponds to the verification of functional incorrectness. Minimality is of course satisfied, as the empty set does not have proper subsets. The property of criticality states that there is a path where no component fails but eventually the hazard occurs. The DCCA formula (on the extended system modeled) is equivalent to the formula $EF\ H$ (on the functional model of the system). This is the negation of the standard property of functional correctness, namely on all paths where no component fails, the hazard will globally not occur (in CTL: $AG\ \neg H$). In other words, if the empty set can be proven to be a critical set, then the system has design errors and is functionally incorrect. The formal proof of this relationship requires properties of conservative extension between the functional system model and the extended system model. In the case study, DCCA shows that the extended system is functionally correct.

Failure Modes and Effects Analysis

The analysis of failure modes $|\Gamma| = 1$ corresponds to traditional Failure Modes and Effects Analysis (FMEA). FMEA analyzes the effects of a component's failure mode on the system in an informal manner. If the failure mode appears to be safety critical, the cause-consequence relationship is noted as one row of an FMEA spreadsheet. If there is a minimal critical singleton set for a hazard H, a correct FMEA must list the hazard H as the effect of the analyzed failure mode. In return, analysis of singleton sets of failure modes with DCCA is a formalization of FMEA. Note that functional correctness is a precondition for formal FMEA. If the system is not functionally correct, there are no minimal critical singleton sets.

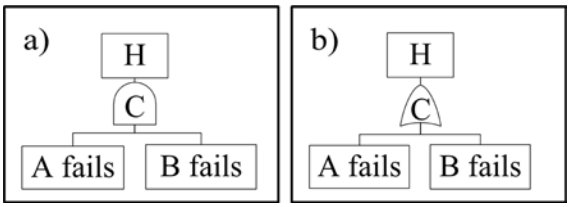
Formal Fault Tree Analysis

Combinations of failure modes, that is, $|\Gamma| > 1$, are often examined by FTA. The main result of FTA is a set of minimal cut sets, which in some way correspond to the minimal critical sets obtained from DCCA. A more precise relationship between minimal cut sets and minimal critical sets can be stated for formal fault trees.

Both informal and formal FTA suffer from the problem that they might produce "cut sets" which are not effective causes but only potential causes of the hazard. Such a "cut set" is in fact not a cut set, as it cannot possibly be responsible for the occurrence of the hazard. We distinguish between optimal and suboptimal fault trees: Optimal fault trees exclusively yield minimal cut sets that are effective causes for the hazard. Suboptimal fault trees, on the other hand, yield some potential causes that are not effective. Theorem 2 holds for a fault tree if it lists all effective causes; it might additionally list some potential causes, hence both optimal and suboptimal fault trees can be proven to be complete. So the ramifications of the additional non-effective causes are not disastrous, but they make the system appear less safe than it actually is. It would therefore be nice to formally decide whether a given fault tree is optimal. However, the effectiveness of a potential cause can only be decided globally for all possible paths of the system. A fault tree, however, considers local cause-consequence relationships only by connecting input events via some gate type to an output event. Due to the modularity of fault trees, it is impossible to derive the effectiveness of some cause from the fault tree itself; the underlying problem is that the effectiveness property is not transitive. For instance, consider a basic event A that is linked to the top event C via an intermediate event B. If the fault tree is proven to be complete, A is a potential cause for B and B is a potential cause for C, and all in all A is a potential cause for C. Assume that A also is an effective cause for B for a set of paths $\vec{\sigma}$ and B is an effective cause for C for a set of paths $\vec{\sigma}'$. Event A might or might not be an effective cause for C, depending on whether $\vec{\sigma} \bigcap \vec{\sigma}' = \varnothing$. If this intersection is empty, there is no path in which A effectively causes B and B effectively causes C, hence an occurrence of A never results in an occurrence of C. This problem can be solved with DCCA, which, as a global technique, is able to check the effectiveness of all causes. In particular, DCCA can be used to check the optimality of a fault tree by testing all minimal cut sets for criticality.

Although both DCCA and FFTA lead to similar completeness theorems, results of both techniques are typically different as DCCA is more precise regarding optimality: In general, FFTA yields weaker cut sets than DCCA. For instance, consider a system with two redundant components A and B, that is, only a failure of both components can lead to the hazard. An intuitively correct fault tree is shown in Figure 17a. But FFTA may miss the opportunity to find this cut set. Instead, a fault tree consisting of a single OR-gate can be proven complete (see Figure 17b) too and thus FFTA yields two singleton minimal cut sets, one for the failure of A and one for the failure of B. This is because the formula

Figure 17. Both fault trees for the same hazard and component failures can be proven complete



$A((\phi_1 \wedge \phi_2)P\psi)$ of the cause-consequence AND-gate in Figure 17a implies the formula $A((\phi_1 \vee \phi_2)P\psi)$ of the C-OR-gate in Figure 17b.

This may make a safe system appear less safe than it actually is. Formally, the relationship between minimal critical sets and minimal cut sets can be states as:

Theorem 4: (Minimal Cut Sets vs. Minimal Critical Sets) Let K be a Kripke structure with failure modes Δ, H be a hazard, and T be a formal fault tree for H. For the set $mcrit(K, \Delta, H)$ of all minimal critical sets and the set $mcut(T)$ of all minimal cut sets the following holds:

$$\forall \Gamma \in mcrit(K, \Delta, H) . \exists \Gamma' \in mcut(T) . \Gamma' \subseteq \Gamma$$

Obviously, Theorem 4 assumes that the same underlying system model is used for both DCCA and FFTA.

The railroad crossing case study exemplifies the discrepancy between minimal critical sets and minimal cut sets. The fault tree shown in the preceding section identifies the failure of the brakes as a single point of failure. The fault tree can be proven complete, therefore giving no indication that it is not optimal. By contrast, DCCA is a global technique that analyzes the system as a whole, which makes it possible to figure out that a failure of the brakes alone is not critical for the considered hazard (although it might be critical when the train is supposed to stop at the next station).

QUANTITATIVE ANALYSIS AND PARAMETER OPTIMIZATION

For real world systems, qualitative assessment of safety properties is not sufficient. It is important to calculate a quantitative estimate of system safety as well. This is typically done by calculating failure probabilities or mean times to or between failures. This section presents an approach for the calculation of these probabilities. The methods shown are used in conjunction with (F)FTA and DCCA, but can also be used in combination with other qualitative safety analysis techniques as well.

Standard Calculation of Probabilities

The standard formula for calculating hazard probabilities from fault trees (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002) is shown below. For a system K and a set of failure modes Δ, the probability P_Γ of a minimal cut/critical set Γ is calculated as the product of the probabilities P_δ of all its elements δ. The probability P_H of a hazard H is calculated as the sum of all its minimal cut sets probabilities:

$$P_\Gamma = \prod_{\delta \in \Gamma} P_\delta$$

$$P_H = \sum_{\Gamma \in mcut(K,\Delta,H)} P_\Gamma$$

These formulae are widely used in engineering and broadly accepted, but they are based on some assumptions about statistical independence. In particular, all failures are assumed to be pair-wise independent. This holds for many applications. If statistical correlation has to be examined, FTA is not a good choice and another approach like common cause analysis or – on the formal side – stochastic model checking has to be used and the probability of the minimal cut sets and hazards have to be calculated separately. The formula above

also neglects second and higher-order terms in the sum, though in practice this does usually not pose a problem as failure probabilities are typically very small.

We stick with the assumption of statistical independence in this section and use the above standard formula as starting point for our extensions. Note that these formulae can be used for DCCA as well by replacing minimal cut sets with minimal critical sets.

Generalizations

The above standard formulae are insufficient for many purposes: They always assume the worst case, that is, all environment inputs are as "bad" as possible. This typically yields too harsh an overestimation. Another deficiency is the use of fixed probabilities for failures. In reality these probabilities are usually not constant, but rather depend on some parameters. To overcome these problems, we generalize quantitative FTA by introducing two new types of probabilities: constraint probabilities and parameterized failure probabilities.

Constraint Probabilities

Most minimal cut/critical sets cause the hazard only if one or more constraints are fulfilled. For example, the malfunction of the brakes is only critical if the train has initiated an emergency stop. While the train is not attempting to decelerate, the failure of the brakes does not have any effect on the system. The qualitative dependency between such constraints, cut sets, and hazards is often integrated in the fault tree with INHIBIT-gates. An INHIBT-gate's enabling condition typically does not describe a failure, but can for example be some environmental influence. Context situations can also be integrated into DCCA by adding the constraint to the set of failure modes. Although some types of FTA respect such dependencies in a qualitative manner, they are in general neglected for quantitative FTA. Therefore, we introduce constraint probabilities for quantitative analysis which reflect such constraining situations. They can be seen as a measure of how likely it is that the inputs from the environment are "bad" enough to make the hazard happen. So we refine the definition of a cut/critical set probability P_Γ to get a better approximation:

$$P_\Gamma = P_\Gamma^c \prod_{\delta \in \Gamma} P_\delta$$

A constraint probability of one, that is, $P_\Gamma^c = 1$, results in the original formula where the environment always behaves as bad as possible. However, if it is possible to estimate P_Γ^c a priori, the results are more precise. The estimate can be approximated by calculating the probabilities of all conditions in INHIBIT-gates along the paths through the tree from the hazard to the elements of the cut sets. An upper bound for the constraint probability is the product of the probabilities of all conditions if statistical independence holds; if not, the maximum is an upper bound for it. These numbers are hard to calculate exactly in practice, hence they are often only approximated. But even if they are not approximated very well, they still can be a great help for safety analysis. This is because varying the constraints allows examining the behavior of the system in different working environments.

Parameterized Probabilities

The second important generalization is that we not only use constant failure probabilities, but allow parameterized probabilities. In many situations the probability of a failure δ (e.g. a relay fails to close) depends on some parameter X (e.g. the spring tension of the relay), therefore we introduce a functional mapping between X and the probability of the failure:

$$P_\delta : Domain(X) \rightarrow [0,1]$$

and write $P_\delta(X)$ for the probability of failure mode δ for some value of X. In principle, there is no restriction on the domain of X, as it only affects which methods are applicable for the solution of the resulting optimization problem. But finite and discrete domains are in general less interesting and rare. In practice, P_δ is usually a continuous probabilistic distribution. If the probabilities depend on more than one parameter, $X := (X_1, ..., X_n)$ is the vector of all of these parameters.

We now substitute all instances of failure probabilities with the corresponding function. As a result, the probability of a cut/critical set is a function of one or more variables. The same applies to the probability of a hazard. Thus, the probabilities of cut/critical sets and hazards are no longer fixed numbers, but rather functions of the free parameters of the system. These functions are called parameterized probabilities:

$$P_\Gamma(X) = \prod_{\delta \in \Gamma} P_\delta(X)$$
$$P_H(X) = \sum_{\Gamma \in mcut(K,\Delta,H)} P_\Gamma(X)$$

Safety Optimization

Parameterized probabilities now allow the combination of quantitative (fault tree) analysis and optimization techniques, which leads to safety optimization. In practice, safety is often a trade-off between different undesired events and the costs to prevent them, i.e., different antagonistic requirements must be balanced. In avionics, for instance, the main goal of a pre-flight safety check on an airplane before takeoff is to make sure that the aircraft is working correctly and will probably not experience any problems during flight. Conversely, an aircraft that is working correctly must not fail the check. Assume that one part of the check is the aberration of the air speed indicator. It is obvious that smaller allowed tolerances increase safety. On the other hand, if acceptable tolerances are too small, many safe aircrafts might fail the pre-flight check and thus may be delayed or cancelled. Obviously, the safest airplane is one that never takes off. Therefore, it is necessary to find an optimal value, balancing the costs of delays and problems during flight. Note, that we do NOT argue for safety problems caused by design flaws which are not fixed for reasons of cost. This approach only addresses hardware failures which ultimately cannot be avoided. For such hardware issues, safety optimization employs mathematical methods to find the "best" compromise between safety and cost in the form of hazard occurrence probabilities and orthogonal objectives which are generally called cost functions.

Cost Function

A cost function describes the total costs of all hazards that the operator of a safety critical system has to expect on average. Costs are estimated by risk assessment for each hazard. It is common practice – even though it may seem unethical – to measure the costs in actual amounts of money; e.g., US railway organizations assume a fixed amount of money for each human casualty.

System operators are mostly interested in the mean costs they have to expect. These costs depend on the probability of the occurrence of a hazard and its absolute costs. In general, a cost function is a mapping from the free parameters of a system into the domain of real numbers. In many cases the cost function consists of two terms: The first one reflects the mean costs $costs(\overline{H}, X)$ associated with the effects of all hazards. The second one directly depends on the cost $costs_P(X)$ of the parameters themselves, e.g., more reliable sensors are more expensive.

$$costs(\overline{H}, X) = \sum_{H \in \overline{H}} (costs_H \ P_H(X))$$
$$costs(X) = costs(\overline{H}, X) + costs_P(X)$$

The costs $costs(\overline{H}, X)$ associated with the hazards are approximated by the weighted sum of hazard probabilities $P_H(X)$ and mean costs $costs_H$ associated with each hazard. Note that probabilities of the hazards are no longer constants – in contrast to standard quantitative FTA – but rather functions of free parameters $X = (X_1, ..., X_n)$. The cost function is a real valued function of the free parameters.

Once the cost function is defined, the design problem of finding good choices for free parameters (i.e. design choices) becomes an optimization problem; the goal is to choose the free parameters X such that the cost function is minimized:

Find $X = (X_1, ..., X_n)$ such that $costs(X)$ is minimal.

To guarantee the existence of the minimum, we restrict the domains to compact intervals. This problem can then be solved with different methods. In simple cases analytical solutions may be found. If the problem is more complex and the cost function is smooth enough (i.e. twice-continuously differentiable), there are a lot of algorithms from the domain of nonlinear programming to solve the problem. The simplest one is the gradient method which finds local minima by calculating gradients iteratively and always following the steepest descent. But there is a wide variety of more elaborate and more efficient algorithms; an introduction to optimization of nonlinear problems may be found in (Luenberger, 1989) and (Nemhauser, Rinnooy Kan, & Todd, 1989).

Application to the Case Study

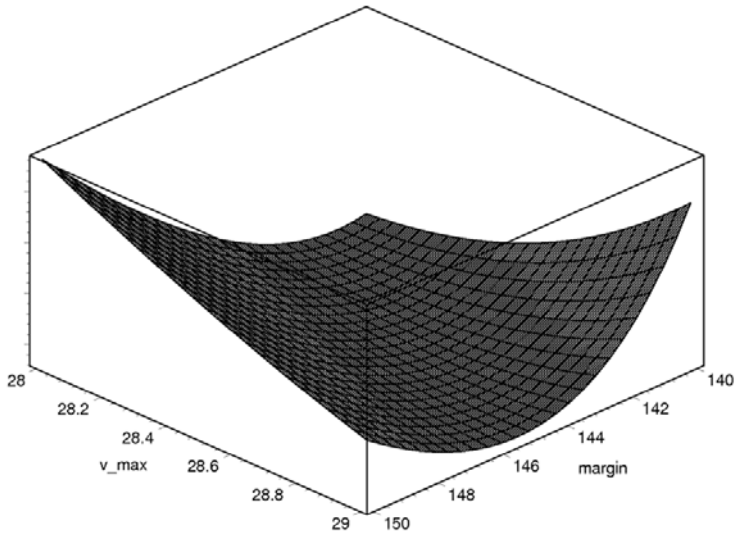
To illustrate the safety optimization approach, we consider two free parameters of the railroad crossing case study: The maximum allowed speed of the train and the safety margin that is added to the calculations of the *Close*, *Query*, and *Stop* positions. The probability of a collision and the estimated time delay for passing the crossing – compared to travelling with maximum speed – are the antagonistic goals considered here. Figure 18 shows the estimated costs as a function of the free parameters. Note that failure probabilities for all considered failures modes have to be specified and the weighting of the antagonistic goals has to be determined. For a detailed description we refer to (Ortmeier, Schellhorn, & Reif, 2004).

If there are only two free variables as in this example and the functions are smooth, solutions may even be found manually by using a 3D plot of the cost function and zooming into it. This method might yield good results by quickly testing a large number of possible combinations of parameter values, even if a specific optimization problem is neither analytically nor numerically solvable. In the example, it turns out that an optimum may be found at a safety margin of 143 m and a maximum speed of slightly more than 28 m/s = 100.8 km/h. This result is somewhat surprising, as safety margins are typically chosen much larger. However, it turns out that increasing the safety margin beyond 150m (in this example) has almost no influence on safety. The reason is that other minimal cut sets (on which the safety margin has no effect) dominate the probability of the hazard. This is also understandable from an intuitive point of view. Safety margins only allow for compensation of deviations of the odometer. A large safety margin has no effects on communication, sensors, and brakes.

Related Work

In this chapter, the case study is modeled by transition systems, more precisely program graphs and Kripke structures, whereas the system properties are formalized as temporal logic formulae verified by model checking. Several alternatives exist: (Arabestani, Bitsch, & Gayen, 2004) use the Uni-

Figure 18. Estimated costs as a function of the free parameters



fied Modeling Language (UML) (OMG, 2010) to model the case study. The UML is a widely used standardized language to model large software systems. Compared to program graphs, it has the advantage of being able to express the structure of the system in addition to the specification of the dynamic behavior. Arabestani et al. first describe the system's structure using UML Class Diagrams to illustrate how the system is organized into components and how the different components interact with each other. They then continue to formalize the behavior of the system using UML State Machines, a more expressive kind of transition system compared to program graphs. Their run-to-completion semantics makes verification more difficult; however, tools have been developed that deal with these complexities. For instance, (Knapp, Merz, & Rauh, Model Checking Timed UML State Machines and Collaborations, 2002) describe the tool HUGO that is able to convert (timed) UML State Machines to the input languages of SPIN or UPPAAL, among others.

Besides these model checking issues, the biggest drawback of an UML-based approach to safety analysis is the lack of a complete standardized formal semantics for the language. In particular, the standard contains so-called "semantic variation points" (OMG, 2010), which allow the developers to adapt some UML constructs to the need of their specific application. (Idani, Ossami, & Boulanger, 2007) discuss some of the issues involved with the use of the UML for the development of EN 50128 certified applications.

(Giese, Tichy, & Schilling, 2004) also use the UML to model the structure of the system; they do, however, not model the system's behavior. Instead, they describe the component structure using UML Component Diagrams in which they also specify the dependencies between components via provided and required interfaces. To assess the safety of the system, they manually derive Boolean logic formulae from the informal component specifications. These formulae describe the failure propagation between components; this is a major difference to our approach, as our explicit modeling of the components' behaviors and their interactions makes the failure propagation information implicit. Because of this manual description of the failure propagation, their approach cannot guarantee that all possible propagations are actually found. Giese et al. combine these failure propagation formulae with a formula describing

the hazard. If the resulting formula can be satisfied, it is possible that the hazard occurs. In contrast to our approach, only Boolean satisfiability checks are required instead of model checking, which makes their approach more scalable than ours. In (Giese & Tichy, 2006), their approach is extended to the analysis of product lines and online-reconfigurable safety-critical systems.

As the UML is widely used in software development, a variety of tools are available to support the design of large software systems. One of these tools that is particularly interesting for safety analysis is developed in the context of the TOPCASED project (Vernadat, et al., 2006), whose purpose is the implementation of an open-source tool to support the development of safety-critical embedded software and hardware systems. It is tightly integrated into the Eclipse Modeling Framework (Steinberg, Budinsky, Paternostro, & Merks, 2008) and builds upon the principle of model-driven engineering as well as formal methods. Systems are modeled in UML notation, amongst others. The TOPCASED project therefore has a more structured and more sophisticated approach to system modeling compared to the program graphs that are used in this chapter. Additionally, it supports automatic code generation and model-to-model transformations as well as simulations of the system behavior. Verification of dynamic properties is supported by translating the system models into the intermediate language Fiacre (Farail, et al., 2008), which is in turn compiled to the input languages of certain model checkers such as the Petri net-based model checker TINA. Fiacre is currently limited to expressing system properties as LTL formulae (Berthomieu, Ribet, & Vernadat, 2004), thus it is not directly possible to perform a DCCA on models developed in the TOPCASED environment.

Petri nets are another kind of transition system that can be translated into equivalent Kripke structures (Latvala & Heljanko, 2000). Thus the techniques presented in this chapter can also be used to evaluate the safety of systems modeled as Petri nets. (Medjoudj & Yim, 2007) present a Petri net model of the case study. They use a custom-tailored tool which allows them to assess the overall safety of the system. The result of their analysis is similar to the results of DCCA, as they obtain a system trace that explains how the system is able to reach a feared state (a hazard in our terminology). This roughly corresponds to the counterexamples generated by NuSMV if a DCCA formula is violated. (Hoenicke & Olderog, 2002), on the other hand, illustrate how a variant of the case study can be modeled using a combination of CSP, Object-Z, and Duration Calculus: CSP is a process algebra that describes the dynamic behavior of communicating sequential processes; Object-Z is a formal specification language based on set theory and first-order predicate logic extended with object-oriented constructs, which is used to describe the structure and operations of the components; and Duration Calculus is a temporal logic in which real-time constraints are expressible. Properties are verified with the help of the UPPAAL model checker. Hoenicke and Olderog do not perform a full analysis of the case study's safety properties, however. Instead, they consider a multi-track version of the case study and analyze certain timing constraints.

Another alternative for modeling safety-critical systems is SCADE (Abdulla, Deneux, Stålmarck, Ågren, & Åkerlund, 2006). Besides the modeling features, it has an integrated model checker that can be used to check safety properties. Furthermore, it features an integrated code generator that is DO-178B-level-A certified. In (Güdemann, Ortmeier, & Reif, 2007) we show how the safety of SCADE models can be assessed with DCCA and thus how DCCA can be employed in conjunction with more widespread modeling tools. In contrast, there is currently no tool support to convert program graphs to any model checker input languages. There is, however, prototypical tool support to automatically generate the DCCA formulae (the tool can be obtained from the authors). In fact, the biggest hindrance to the applicability of DCCA

to real-world systems – as well as any technique based on model checking – concerns the state space explosion problem; real-world systems quickly become too complex to be model checked within reasonable time and memory constraints. There are several ways to alleviate the consequences: For one, faster hardware is able to cope with larger models. In this area, GPU-based model checking (Edelkamp & Sulewski, 2010) seems particularly promising with speed-ups by at least one order of magnitude compared to traditional CPUs. Another promising approach is counterexample-guided abstraction refinement (Clarke, Grumberg, Jha, Lu, & Veith, 2000), where seemingly irrelevant parts of the model (for the verification of a certain property) are abstracted automatically, therefore having the potential of significantly reducing the number of states that have to be explored. It can be shown that all formulae valid in the abstract model also hold in the concrete one. If the abstract model produces a counterexample, either the concrete model also does not satisfy the property, or the counterexample is spurious and is automatically analyzed to construct a better abstraction function. However, at some point models of real-world systems just have too many states to be model checked efficiently and none of the aforementioned countermeasures are able solve the underlying theoretical problem; namely, the exponential growth of system models. Still, we plan on experimenting with GPU-based model checkers in the future and will also look for ways to combine counterexample-guided abstraction refinement and DCCA.

The FSAP-NuSMV/SA platform (Bozzano & Villafiorita, 2007) is another tool for the design and analysis of complex safety-critical systems. Even though the tool provides a graphical user interface, the system model is specified textually in the input language of NuSMV. In exchange, the tool automatically augments the system model with failure modes and provides a pattern-based definition of temporal logic formulae to express safety properties. Property verification is performed by the NuSMV model checker. The platform can additionally perform an ordering analysis which determines the order of safety-critical events that must be fulfilled for the hazard to occur. DCCA as presented in this chapter does not provide this information, possibly yielding results that are too pessimistic. However, it is possible to consider temporal ordering with DCCA as outlined in (Güdemann, Ortmeier, & Reif, 2008). Fault Tree Analysis can also be extended to cope with the ordering of events: (Coppit, Sullivan, & Dugan, 2000) define the formal semantics of Dynamic Fault Trees which also include constructs to model functional dependencies. As Dynamic Fault Trees do not assume statistical independence of the failure modes, the results of the quantitative analysis better reflect the actual system. (Codetta-Raiteri, 2005) presents graph transformation rules to convert dynamic fault trees to a special kind of stochastic Petri net, on which the safety analysis is subsequently performed.

SLIM (Bozzano, Cimatti, Roveri, Katoen, Nguyen, & Noll, 2009) is a component-based modeling language which supports both discrete data as well as continuous evaluations of time and variables whose values are determined by differential equations. Erroneous behavior and its propagation through the subcomponents of a system, degraded modes of operation, and error recovery are all expressible in the language. Additionally, failure modes modeled in SLIM can each be annotated with a probability that indicates their occurrence likeliness within a given unit of time. The semantics of SLIM are formally defined; properties are expressed using CTL, LTL, or Probabilistic CTL, among others. Non-probabilistic properties are checked using the NuSMV model checker, whereas the MRMC probabilistic model checker is used for the quantitative analysis of the model (Bozzano, Cimatti, Katoen, Nguyen, Noll, & Roveri, 2009). Probabilistic model checking yields more precise results than the estimates produced by our approach, as it does not assume statistical independence of the failure modes.

We are therefore currently investigating the use of probabilistic model checking to improve our techniques.

CONCLUSION

The processes, tools, and techniques used for the development of safety-critical systems are often determined by industry standards such as IEC 61508, EN 50128, or DO-178B. In recent years, these standards began to highly recommend the use of formal methods throughout the entire development process. With the help of formal methods, the structure and behavior of a system can be described in a mathematically precise manner, making it possible to verify (safety-related) properties of the system in conjunction with various tools and analysis techniques. Formal methods are usually employed in addition to traditional, informal development techniques. Failure Modes and Effects Analysis or Fault Tree Analysis remain important analysis methods in the realm of safety-critical systems, because they support the identification of relevant and possible hazards and failure modes. Model-based analysis techniques, on the other hand, are generally focused on finding relationships between already identified hazards and failures.

The ForMoSA approach combines traditional safety analysis techniques, formal methods, and mathematical optimization to increase the overall precision and validity of the results of the safety assessment process. While traditional techniques provide first preliminary results, formal specification of a system's structure, behavior, and properties often already yields new insights that show omissions or inconsistencies in the original informal system specification. Hence, many problems are identified earlier in the development process, decreasing the effort and costs to fix them. Subsequently, occurrence patterns and effects of component failures and other faults are thoroughly considered and integrated into the system model. This integration is guaranteed to preserve the system's original desired and intended behavior as long as the rules of conservative integration are followed.

Formal Fault Tree Analysis and Deductive Cause Consequence Analysis are both based upon a formal understanding of cause-consequence relationships. The notion of potential and effective causes distinguishes between events that only might result in a consequence and events that demonstrably cause a consequence. As traditional Fault Tree Analysis, FFTA generally yields only potential causes for a system hazard. DCCA, on the other hand, has a global view on the system and can therefore also analyze the effectiveness of all identified potential causes. The advantage of FFTA over DCCA concerns infinite state systems, as they can be analyzed with FFTA with the help of our interactive theorem prover KIV. Theoretically, a DCCA could also be conducted using a theorem prover that supports CTL, but as it is a global technique, proofs are complicated and time-consuming.

Both techniques yield sets of (combinations of) component failures called minimal cut sets and minimal critical sets, which describe potential or effective causes for the hazard under consideration. Even though minimal cut sets do not necessarily describe effective causes, they share the most important property with minimal critical sets: If for each identified potential cause only one component failure is either avoided entirely or at least made less likely, the hazard cannot occur at all or at least its likeliness decreases accordingly. Thus, both analysis techniques are complete in the sense that no relevant causes are neglected. The results obtained from FFTA and DCCA can then be used to improve the system's safety. For example, redundant components might be introduced to decrease the risk of a hazard. In the case that the system is functionally incorrect, that is, if safety issues manifest themselves even in the absence of any component failures, larger design changes might be necessary to reach the required safety

level. However, whether such a possibly costly change is made also depends on the likelihood that a safety-relevant problem occurs while the system is in use.

The ForMoSA approach also covers such quantitative aspects, taking free system parameters into account by specifying probability distributions for component failures as a function of these parameters. As there are typically many different hazards and functional goals for a system, the values chosen for these parameters have to be carefully balanced in order to find an optimal compromise for all relevant (safety) goals of the system.

All ForMoSA techniques have been successfully applied to several case studies, one of which is the radio-based railroad crossing. A simplified version of this case study is presented in this chapter to illustrate the applicability of the approach without considering all the complexities of the real-world system. In the literature, the case study is modeled and analyzed with various other kinds of specification mechanism and verification techniques, each having its distinct advantages and disadvantages. Due to their reliance on model checking, the techniques presented in this chapter have the downside that their applicability for large real-world systems is limited. However, if they can be used, they guarantee that all effective causes for a hazard are found; provided that all relevant component failures are modeled in the first place.

ACKNOWLEDGMENT

This work has partly been funded by the Deutsche Forschungsgemeinschaft priority program "Integration of Software Specification Techniques for Applications in Engineering". We would like to express our gratitude to Alexander Knapp for his invaluable input on the definition of program graphs and his helpful remarks and comments on the presentation of the topics in this chapter. We also thank Matthias Tichy and Florian Nafz for their time and their insightful comments. Many thanks also go to the reviewers of this chapter who pointed out weaknesses and shortcomings and thus helped us to improve the overall understandability.

REFERENCES

Abdulla, P., Deneux, J., Stålmarck, G., Ågren, H., & Åkerlund, O. (2006). Designing Safe, Reliable Systems Using Scade. In Margaria, T., & Steffen, B. (Eds.), *Leveraging Applications of Formal Methods* (pp. 115–129). Berlin, Heidelberg: Springer Verlag. doi:10.1007/11925040_8

Abrial, J.-R. (2007). Theory Becoming Practice. In *Journal of Universal Computer Science* (pp. 619–628). Formal Methods.

Arabestani, S., Bitsch, F., & Gayen, J.-T. (2004). Precise Definition of the Single-Track Level Crossing in Radio-Based Operation in UML Notation and Specification of Safety Requirements. In *H. Ehrig, W. Damm, J. Desel, M. Große-Rhode, W. Reif, E. Schnieder, et al., Integration of Software Specification Techniques for Applications in Engineering* (pp. 119–144). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-27863-4_9

Baier, C., & Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.

Balser, M., Reif, W., Schellhorn, G., Stenzel, K., & Thums, A. (2000). *Formal system development with KIV. Fundamental Approaches to Software Engineering*. Springer.

Ben-Ari, M. (2008). *Principles of the Spin Model Checker*. Springer Verlag.

Bengtsson, J., & Yi, W. (2004). Timed Automata: Semantics, Algorithms and Tools. In Rozenberg, R. W., & G., (eds) *Lecture Notes on Concurrency and Petri Nets.* Springer-Verlag.

Berthomieu, B., Ribet, P.-O., & Vernadat, F. (2004). The tool TINA - Construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research.*

Bitsch, F. (2001). Safety Patterns - The Key to Formal Specification of Safety Requirements. *Proceedings of the 20th International Conference on Computer* (pp. 176-189). Berlin / Heidelberg: Springer Verlag.

Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., & Roveri, M. (2009). The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In Buth, B., Rabe, G., & Seyfarth, T. (Eds.), *Computer Safety, Reliability, and Security* (pp. 173–186). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-642-04468-7_15

Bozzano, M., Cimatti, A., Roveri, M., Katoen, J.-P., Nguyen, V. Y., & Noll, T. (2009). Codesign of dependable systems: A component-based modeling language. In *7th IEEE/ACM International Conference on Formal Methods and Models for Co-Design, MEMOCODE* (pp. 121 -130). IEEE.

Bozzano, M., & Villafiorita, A. (2007). The FSAP/NuSMV-SA Safety Analysis Platform. In *International Journal on Software Tools for Technology Transfer (STTT)* (pp. 5–24). Berlin, Heidelberg: Springer Verlag.

Bozzano, M., & Villafiorita, A. (2010). *Design and Safety Assessment of Critical Systems.* CRC Press (Taylor and Francis), an Auerbach Book.

Bruns, G., & Anderson, S. (1993). Validating Safety Models with Fault Trees. In J. Górski, *SafeComp'93: 12th International Conference on Computer Safety, Reliability, and Security* (pp. 21-30). Springer-Verlag.

CENELEC. (2011). *Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems (EN 50128:2011).*

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking. *Proc. International Conference on Computer-Aided Verification (CAV 2002).* Copenhagen, Denmark: Springer.

Clarke, E., Grumberg, O., Jha, S., Lu, Y., & Veith, H. (2000). Counterexample-Guided Abstraction Refinement. In Emerson, E., & Sistla, A. (Eds.), *Computer Aided Verification* (pp. 154–169). Berlin, Heidelberg: Springer Verlag. doi:10.1007/10722167_15

Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., & Meseguer, J. (2007). *All About Maude - A High-Performance Logical Framework*. Springer Verlag.

Codetta-Raiteri, D. (2005). The Conversion of Dynamic Fault Trees to Stochastic Petri Nets, as a case of Graph Transformation. In *Electronic Notes Theoretical Computer Science* (pp. 45-60).

Coppit, D., Sullivan, K., & Dugan, J. (2000). Formal Semantics of Models for Computational Engineering: A Case Study on Dynamic Fault Trees. In *International Symposium on Software Reliability Engineering.* IEEE.

Damm, W., Josko, B., Hungar, H., & Pnueli, A. (1998). A Compositional Real-time Semantics of STATEMATE Designs. *COMPOS'97, Volume 1536 of LNCS* (pp. 186-238). Berlin / Heidelberg: Springer.

Edelkamp, S., & Sulewski, D. (2010). Efficient explicit-state model checking on general purpose graphics processors. In *Proceedings of the 17th international SPIN conference on Model checking software* (pp. 106-123). Enschede, The Netherlands: Springer Verlag.

Farail, P., Gaufillet, P., Peres, F., Bodeveix, J.-P., Filali, M., Berthomieu, B., et al. (2008). FIACRE: an intermediate language for model verification in the TOPCASED environment. *European Congress on Embedded Real-Time Software (ERTS)*. SEE.

Fenelon, P., McDermid, J., Nicholson, A., & Pumfrey, D. (1995). Experience with the application of HAZOP to computer-based systems. *Proceedings of the 10th Annual Conference on Computer Assurance.* Gaithersburg, MD: IEEE.

Giese, H., & Tichy, M. (2006). Component-Based Hazard Analysis: Optimal Designs, Product Lines, and Online-Reconfiguration. In Górski, J., *Proc. of the 25th International Conference on Computer Safety, Security and Reliability (SAFECOMP)* (pp. 156-169). Gdansk, Poland: Springer Verlag.

Giese, H., Tichy, M., & Schilling, D. (2004). Compositional Hazard Analysis of UML Components and Deployment Models. In M. Heisel, P. Liggesmeyer, & S. Wittmann, *Proc. of the 23rd International Conference on Computer Safety, Reliability and Security (SAFECOMP)* (pp. 166-179). Potsdam, Germany: Springer Verlag.

Górski, J. (1994). *Extending safety analysis techniques with formal semantics. Technology and Assessment of Safety Ciritcal Systems* (pp. 147–163). London: Springer Verlag.

Güdemann, M., Ortmeier, F., & Reif, W. (2007). Using Deductive Cause Consequence Analysis (DCCA) with SCADE. *Proceedings of SAFECOMP 2007*. Springer LNCS 4680.

Güdemann, M., Ortmeier, F., & Reif, W. (2008). Computing Ordered Minimal Critical Sets. *Proceedings of the 7th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 08).*

Hänsel, F., Poliak, J., Slovák, R., & Schnieder, E. (2004). Reference Case Study "Traffic Control Systems" for Comparison and Validation of Formal Specifications Using a Railway Model Demonstrator. In Ehrig, H., Damm, W., Desel, J., Große-Rhode, M., Reif, W., & Schnieder, E. (Eds.), *Integration of Software Specification Techniques for Applications in Engineering* (pp. 96–118). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-27863-4_8

Hansen, K., Ravn, A., & Stavridou, V. (1998). From safety analysis to software requirements. *IEEE Transactions on Software Engineering*, *24*(7), 573–584. doi:10.1109/32.708570

Hansen, K. M., Ravn, A. P., & Stavridou, V. (1994). *From safety analysis to formal specification. ProCoS II document*. Technical University of Denmark.

Harel, D., & Naamad, A. (1996). *The STATEMATE semantics of statecharts. Transactions on Software Engineering and Methodology* (pp. 293–333). New York, NY, USA: ACM.

Havelund, K., Lowry, M., Pecheur, C., Penix, J., Visser, W., & White, J. (2000). Formal Analysis of the Remote Agent Before and After Flight. In *The Fifth NASA Langley Formal Methods Workshop*. Virginia.

Haxthausen, A. E. (2010). *An Introduction to Formal Methods for the Development of Safety-critical Applications. Kgs*. Denmark: Lyngby.

Hoare, C. (1985). *Communicating Sequential Processes*. Prentice Hall.

Hoenicke, J., & Olderog, E.-R. (2002). Combining Specification Techniques for Processes Data and Time. In M. Butler, L. Petre, & K. Sere, *Integrated Formal Methods* (pp. 245-266). Springer Verlag.

Idani, A., Ossami, D.-D., & Boulanger, J.-L. (2007). Commandments of UML for Safety. In *International Conference on Software Engineering Advances, ICSEA* (p. 58). IEEE.

IEC. (2010). *Functional safety of electrical/electronic/programmable electronic safety-related systems (IEC 61508)*.

Knapp, A. (2004). *Semantics of UML State Machines*. München: Ludwig-Maximilians-Universität, Technical Report 0408.

Knapp, A., Merz, S., & Rauh, C. (2002). Model Checking Timed UML State Machines and Collaborations. In W. Damm, & E. Olderog, *Proc. 7th Int. Symp. Formal Techniques in Real-Time and Fault Tolerant Systems* (pp. 395-416). Berlin: Springer Verlag.

Kwiatkowska, M., Norman, G., & Parker, D. (2011). PRISM 4.0: Verification of Probabilistic Real-time Systems (to appear). In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*. Springer Verlag.

Latvala, T., & Heljanko, K. (2000). Coping with Strong Fairness. In *Fundamenta Informaticae* (pp. 175–193). IOS Press.

Leitsch, R. (1995). *Reliability Analysis for Engineers: An Introduction*. Oxford Science Publications.

Leveson, N. (1995). *Safeware: System Safety and Computers*. Addison-Wesley Publishing.

Leveson, N. (2002). *A new approach to system safety engineering*. Aeronautics and Astronautics Massachusetts Institute of Technology.

Luenberger, D. (1989). *Linear and nonlinear programming*. Addison-Wesley Publishing.

Medjoudj, M., & Yim, P. (2007). Extraction of Critical Scenarios in a Railway Level Crossing Control System. *International Journal of Computers, Communications & Control*, *n.d.*, 252–268.

Nemhauser, G., Rinnooy Kan, A., & Todd, M. (1989). *Optimization*. ElsevierScience Publishers B.V.

OMG. (2010). *Unified Modeling Language Specification formal/2010-05-03.* Version 2.3.

Ortmeier, F., Schellhorn, G., & Reif, W. (2004). Safety Optimization of a Radio-Based Railroad Crossing. In E. Schnieder, & G. Tarnai, *Formal Methods for Automation and Safety in Railway and Automotive Systems.* Braunschweig.

Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*. NJ, USA: Prentice Hall PTR.

Reifer, D. (1979). Software failure modes and effects analysis. *IEEE Transactions on Reliability*, *n.d.*, 147–249.

RTCA. (1992). *Software Considerations in Airborne Systems and Equipment Certification (DO-178B).*

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF: Eclipse Modeling Framework*. Addison-Wesley Professional.

Storey, N. (1996). *Safety-Critical Computer Systems*. Addison Wesley.

Vernadat, F., Percebois, C., Farail, P., Vingerhoeds, R., Rossignol, A., Talpin, J., et al. (2006). The TOPCASED Project - A Toolkit in OPen-source for Critical Applications and System Development. In *Data Systems In Aerospace (DASIA)*. Berlin: European Space Agency (ESA Publications).

Vesley, W., Dugan, J., Fragola, J., Minarick, J., & Railsback, J. (2002). *Fault Tree Handbook with Aerospace Applications*. Washington, D.C.

Woodcock, J., Larsen, P. G., Bicarregui, J., & Fitzgerald, J. (2009). Formal methods: Practice and experience. In *ACM Computation Survey* (pp. 1-36).

KEY TERMS AND DEFINITIONS

Deductive Cause Consequence Analysis: A safety analysis technique based on model checking that identifies the minimal cut sets for a given hazard of a given system.

Extended System Model: Formal description and representation of the system under consideration including its internal structure and its functional as well as erroneous behavior.

Fault Tree Analysis: A well-established safety analysis technique where a safety engineer recursively analyzes the causes of a hazard.

Formal Methods of Software Engineering: Mathematical techniques for the specification, modeling, and verification of software and hardware systems.

Hazard: A system state that has the potential of causing damage to people or the environment.

Minimal Cut Sets: Combinations of component failures that might cause a hazard.

Model Checking: An automated technique for checking whether a system model meets a given specification.

Safety Optimization: The act of balancing (antagonistic) safety parameters or safety goals in order to obtain a parameter set that is considered "best" by some measure.

System Specification: Formal description of the externally visible behavior (properties) of a system.

Temporal Logics: A mathematical specification formalism to express system properties in relation to time.

Section 3

Verification and Validation

Chapter 5
Verification and Validation of Interoperability

Lars Ebrecht
DLR (German Aerospace Center), Institute of Transportation Systems, Germany

Michael Meyer zu Hörste
DLR (German Aerospace Center), Institute of Transportation Systems, Germany

ABSTRACT

The chapter shows an approach to use existing test methods to prove technical as well as operational interoperability. The first kinds of tests are test sequences to validate conformity of a single constituent – here, an on-board on-board unit (OBU) of the European Train Control System (ETCS) in the European Rail Traffic Management System (ERTMS). The second kind of tests is the integration test for assemblies – here, the complete on-board equipment. The third kinds of tests are the tests for the validation of operational serviceability. An approach for the stepwise integration of the different kinds of tests is shown. As a conclusion the perspective for the use of these test sequences in an independent test lab is given.

INTRODUCTION

Trains need long distances for braking. A regular train needs approx. 1 km for braking to standstill from 160 km/h. So they cannot be driven on sight as road vehicles. Therefore train control systems are in use since many years. Those systems supervise the driving speed of the trains and trigger an automatic braking if the driver does not react properly. These train control systems have been subject of national regulation for many years. Since the year 1989 the European Commission supports the specification and implementation of a harmonized European System in order to substitute the more than 20 proprietary national signaling systems by one harmonized system. This shall improve trans-border railway operation and open the markets of railway operation as well as train control systems.

The European Rail Traffic Management System (ERTMS) consists out of the safe communication System GSM-Railway (GSM-R) and the

DOI: 10.4018/978-1-4666-1643-1.ch005

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

European Train Control System (ETCS). ERTMS/ ETCS should implement technical as well as operational interoperability on the trans-European railway network. This means that a train, which is equipped with ETCS, can run on every line, which is equipped with ETCS.

Technically seen the ETCS on-board unit (OBU) consists of a central vital computer system, a spot transmission system from trackside transponders called EuroBalises, the wireless communication system EuroRadio, one or two Driver-Machine-Interfaces (DMI), a Juridical Recording Unit (JRU) and a train interface unit (TIU). In the ETCS application level 1, Euro-Balises or short Balises are used to transmit the permission to run, given by a conventional wayside signal, to the train. In the ETCS application levels 2 and 3, movement authorities are sent by the so-called radio block center (RBC) to the train and shown on-board by the DMI to the driver.

Six different companies commit themselves to implement ETCS and to provide products to the railways in Europe. This leads to the need to prove that all the products fulfill the following high-level requirements:

1. They have been implemented according to the European system requirement specification – the so-called conformity,
2. They interact technically in the correct way – the so-called technical interoperability,
3. They fulfill together the operational functionality – the so-called operational interoperability,
4. They are doing what the railway needs to perform the operational tasks – the so-called serviceability
5. They are doing all this under all conditions in a safe way.

So those components for safe railway applications need to be tested comprehensively before taken into operation. These tests have to

BACKGROUND

Currently ETCS is getting more and more into operation in Europe (Stanley2011)..

Some European countries are already operating ETCS (see Figure 1), for example, Switzerland uses ETCS Level 2 on the routes Mattstetten - Rothrist and Lötschberg - Base tunnel Spain successfully installed ETCS Level 1 and Level 2 on the route Madrid - Barcelona, and the Deutsche Bahn AG (Germany Railways) is preparing several routes (French boarder - Saarbrücken - Ludwigshafen (POS north), Nürnberg - Ingolstadt - Munich (NIM) satisfying the requirements of the current legal ETCS system requirement specification (SRS v2.3.0d).

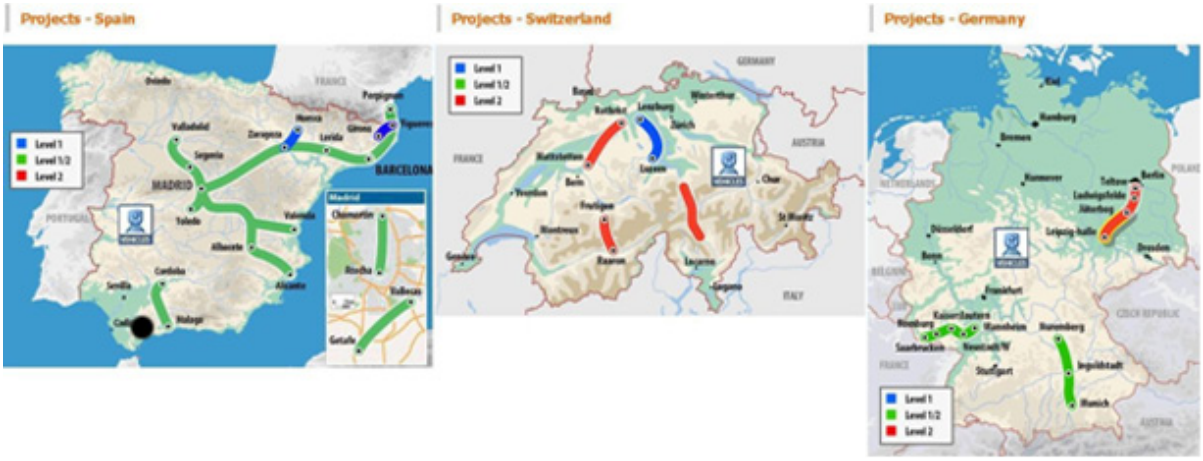
In order to validate the interoperability of ETCS components they could be tested on a real track but the costs and efforts are very expensive (e.g., a train has to be moved to the track, the trips have to be arranged with daily traffic, etc.). To save costs it is necessary as a first step to check the conformity and interoperability of new ETCS components in a laboratory as well as to check if they are in line with the requirements of the ETCS.

Figure 2 illustrates the proposed stepwise methodology and approach for testing technical, line-specific and operational requirements and the interaction and collaboration of the different components.

TECHNICAL INTEROPERABILITY

The ETCS onboard units (OBU) will be tested on component level using the conformity and interoperability Test Standard for ETCS onboard units (Subset-076 (UNISIG, 2008])) and Reference Test Architecture (Subset-094 (UNISIG, 2009)) released by the European Railway Agency with the Test Sequence Debugger (TSD) in order to approve the compliancy and conformity against the SRS. The technical interoperability of train- and trackside components, i.e. OBU and Radio Block

Figure 1. Example of ongoing ETCS projects in Europe (UNI)



Centre (RBC), will be checked also in a laboratory on the integration level. At least operational line-specific tests can show the feasibility and suitability on system level including all parts of the entire system also in a laboratory. The OBU and ETCS conformity approval on component level and its use through the different layers will ensure the compliancy and consistency between the operational of the Railway Undertaking and the technical requirements of the SRS. With the

Figure 2. Classification Scheme, from Lab Test to Field Test

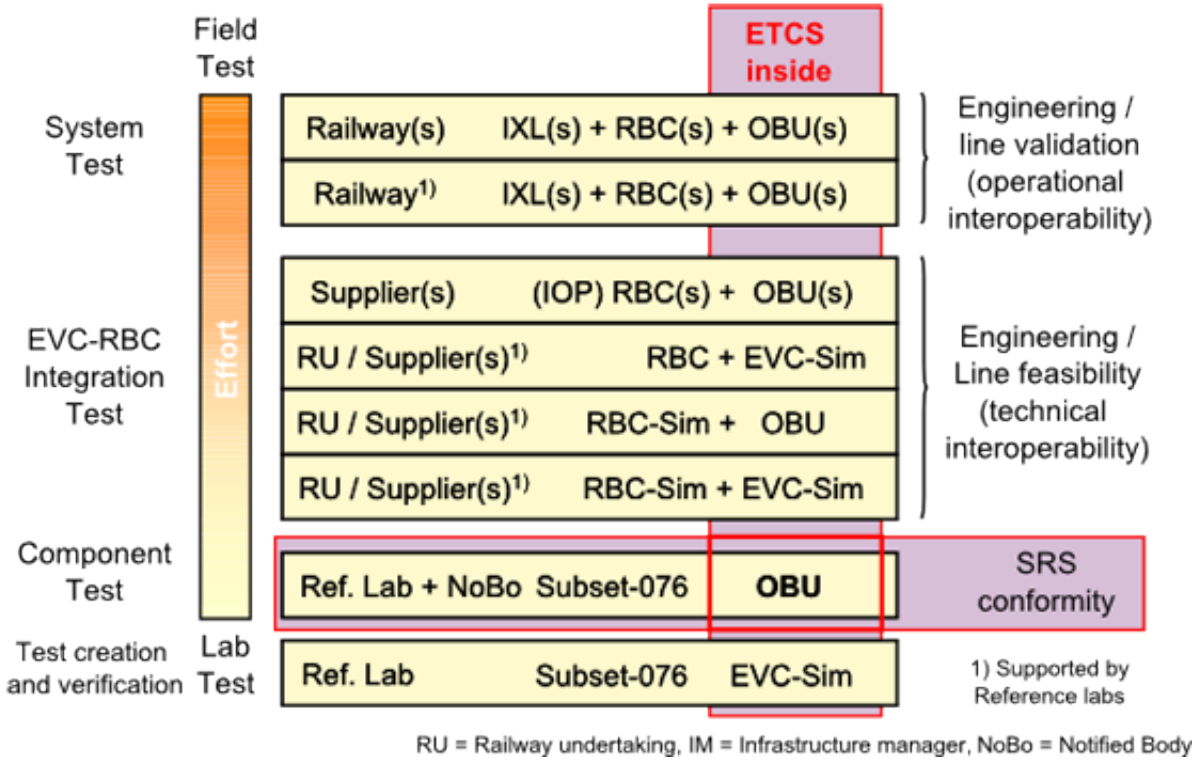
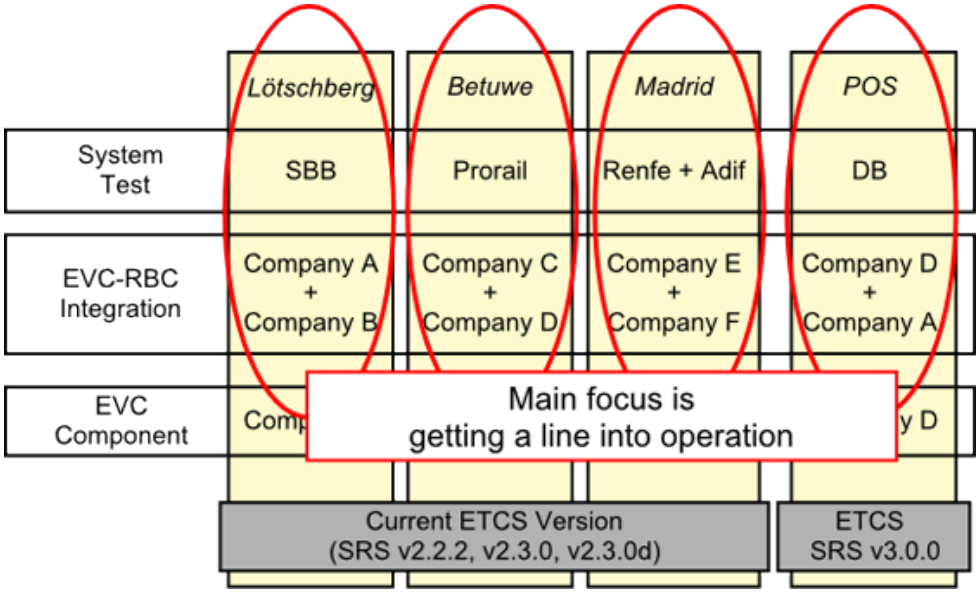


Figure 3. Interoperability for each line



different levels having the ETCS conformity as base, it is granted that ETCS is inside all the other level above (See Figure 2, OBU and EVC-Sim as caliber for higher levels. The EVC-Sim is a functional simulator of the OBU where EVC is standing for European Vital Computer, which is the hardware of the OBU).

Figure 3 shows different suppliers participating in equipping a track. This figure focuses on the GSM-R interface between train and track (e.g., "Supplier B" delivers the RBC, "Supplier A" delivers the OBUs). Of course, there are many more Subsystems in ETCS, which have to be SRS conform to reach real interoperability, but the OBU-RBC interface is the most complex one.

The first target for the Railway Undertaking is to get the line into operation therefore a system test is done line by line (shown by the red boxes enframing the single lines). Even though each line can be part of corridor across Europe there is no integration test between those lines.

In these tests is shown for example that the following functionalities are working technically properly:

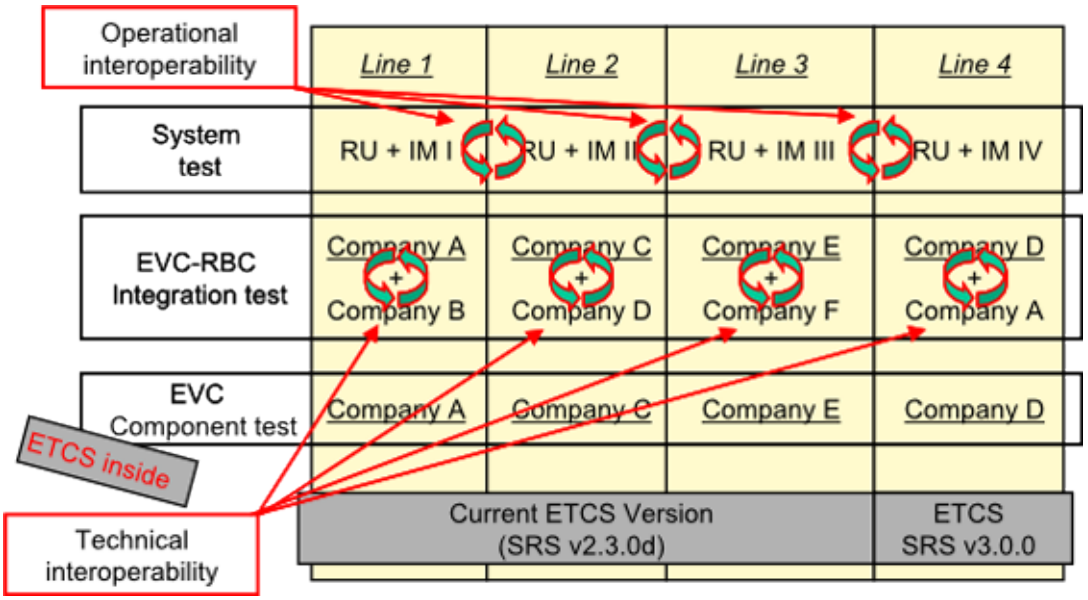
- The content of the Balises is received and decoded correct
- Radio telegrams received and decoded correct
- The OBU reacts correct on requests
- All radio messages and Balise telegrams are recorded in the JRU
- The braking functions triggers the TIU etc.

OPERATIONAL INTEROPERABILITY

Technical Interoperability can be proven between different lines by an OBU conformity test, however there is no test for operational interoperability.

Figure 4 shows the technical interoperability brought by the OBU conformity test and the operational interoperability in the system layer. To take advantage of the basic idea of ETCS, using one Train Control System to cross Europe, a common set of operational rules is needed. This common set has to harmonize the operational rules of the different countries respectively the Railway Undertakings and the Infrastructure

Figure 4. Focus interoperability on system layer



Managers. This common set allows line independent operation.

Furthermore this picture illustrates the effect by introducing the new SRS Baseline 3 (e.g. SRS v 3.0.0). There will be lines running different SRS versions. This will necessitate further compatibility checks between the different major SRS versions. Of course only the OBUs will be affected by this.

There are 3 main interoperability parts, first the technical interoperability, proven by OBU conformity test, second the operational interoperability and third the SRS Version compatibility.

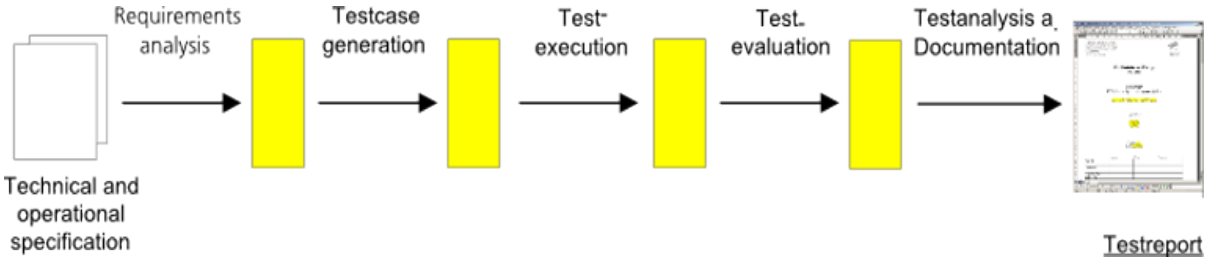
The operational interoperability can be laboratory-confirmed by expanding existing test methods of the Subset 076 conformity test.

In these tests is shown for example that the following functionalities are working technically properly:

- The movement authority shown on the DMI is correct with respect to the signal aspect
- The relation between radio telegrams and line layout is correct
- The Balises are giving the right data with respect to the line layout
- Braking distances are correct
- Shunting areas are correct coded in the Balises etc.

An example of the concrete implementation process shall be given here: When supplier A has finished the implementation of his OBU, it is shipped to an independent lab. There is the conformity test according to the ETCS subset 076 performed. After finishing successfully the conformity test the notified body issues the declaration of conformity. Now the OBU undergoes the technical interoperability tests. There are different possibilities to do this. On of them is the so-called IOP-testing: the OBU is connected to the lab of the supplier A, while the RBC which is the trackside equipment for the relevant line, is in the lab of supplier B. Now the interoperability tests are run remotely. Finally the operational serviceability is typically tested in field tests before the permission

Figure 5. General test process for SS076 conformity



for the start of operation is given by the national railway safety authority.

Due to the high relevance for the certification a validation and verification procedure is required for the operational interoperability test specifications. They are normally specified from operational test specialist by using the national requirement specification and the national operational rules. They have to be defined by using the conformity test cases and adding the missing national operational test cases. They have to be reviewed by test specialists as well as by operational specialists. In the next step they can be validated in an independent lab by using the official test sequence debugging tool, which is published by the European Railway Agency (ERA). This normally leads to the detection of failures and mistakes which have not been detected in the specialists review.

Relation to Safety Testing

The full functional testing approach discussed here, intends to show that a specific functionality has been realized by the system under test. This is not a direct safety test, but has got some implications towards safety. It can lead to hazardous situations if a system is not fulfilling the specified functionality in the correct way. Due to that reason the conformity as well as the interoperability are related to the safety and have to be shown in the certification process. These tests are using normally realistic or operationally typical values.

The safety tests are defined independent in the process given by the standards for the development of safety-relevant electronic systems. These tests are typically tests of border values or failures. This approach is used to show, that the systems behaves in safe way, even when a failure happens.

The Testing Methodology for Subset 076 Conformity Test

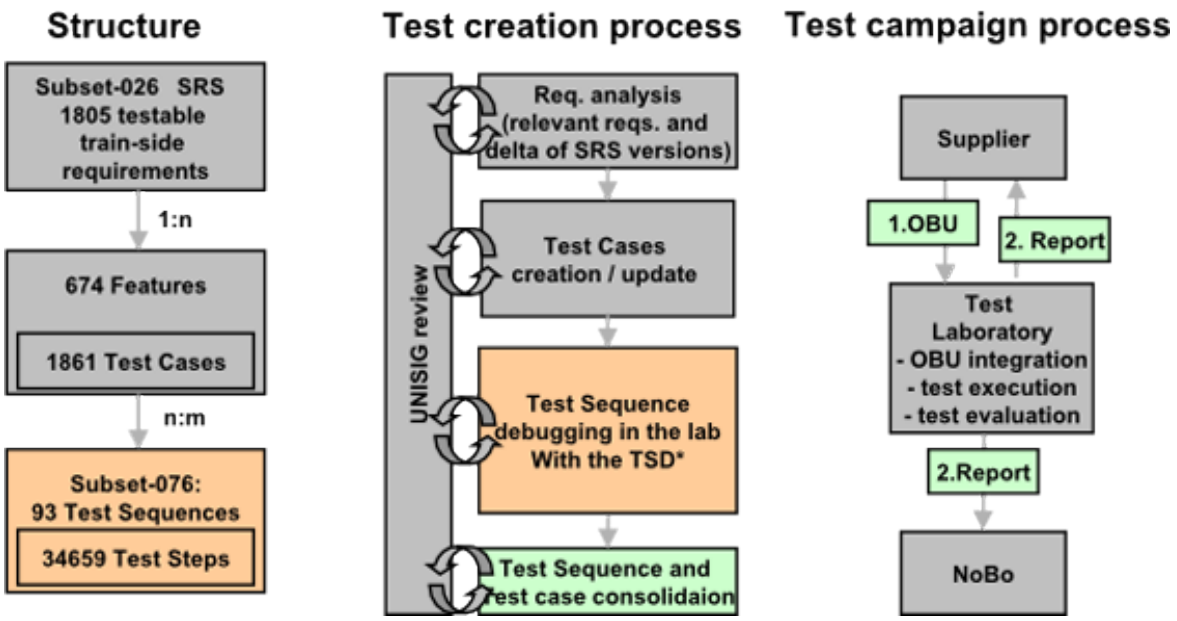
The next paragraph explains the current test process for OBU conformity tests.

The general test process is shown in Figure 5. As a first step the specification needs to be analysed and requirements have to be defined. Those requirements have to be composed to test cases. This step is one of the most important in the overall Test Process, because the requirements are only available in human readable format and have to be transferred into a format readable by the test-bench. Based on the test cases the test can be executed and evaluated. The evaluation result will be analyzed and summarized in a test report.

Transferring the requirements into test cases happens manually. All other steps can be executed in an automated way.

Figure 6 shows the "Subset 076" test creation process concerning conformity tests. All steps during the creation of the test sequences are done in a close cooperation between the independent test labs and the UNISIG suppliers. The left column "Structure" shows the process from the specification to the test sequence. First all train-side requirements, described by the SRS (System Requirement

Figure 6. Subset 076 - OBU conformity test process



Specification, e.g. UNISIG (SRS), 2008) have to be identified. Because the SRS is written as human readable text and not as an algorithm all requirements have to be transferred into features manually. I.e. the features are the "machine readable" requirements. A feature consists of several test cases, which are testing related requirements with different constraints. Further every test case consists of several test steps. The test steps define all inputs and outputs on the respective interface. To ensure testing of all requirements all test cases have to be tested. Thus the test cases have been strung together to Test Sequences. All train-side requirements and furthermore the conformity to the specification can be checked by running a total of 93 test sequences.

The test creation process, shown in the middle column, is done by having a close collaboration with the UNISIG group. This is necessary to ensure a common target. Because every real OBU, which is the System under Test (SUT), has slightly different characteristics and the test cases are meant to be valid for OBUs of all suppliers, the suppliers can collaborate in the development of the test cases and test sequences. Furthermore this collaboration increases the practical supply of the Test Execution. To find deviations and not testable scenarios in the test sequences a special software caliber, the TSD is used. The TSD is a pure software implementation of the SRS. All test sequences are tested by the independent test labs with the TSD. Found Deviations are discussed between the UNISIG Group, the independent test labs and the Software-Company which developed the TSD. For all found issues a final decision is made and the test sequences are transferred into a consolidated version.

A test campaign with a real OBU can also be executed by the test lab. The test lab integrates the OBU into the lab infrastructure and creates a test report after running all test sequences. The Testreport can help the supplier to be certificated by a Notified Body (NoBo), depending on the test lab a collaboration with NoBo during the test phase is possible.

Figure 7. Top left: Mockup for RailSiTe® Simulation, Top right: RailSiTe® Simulation Core, Bottom: Track Visualisation



Testing Environment

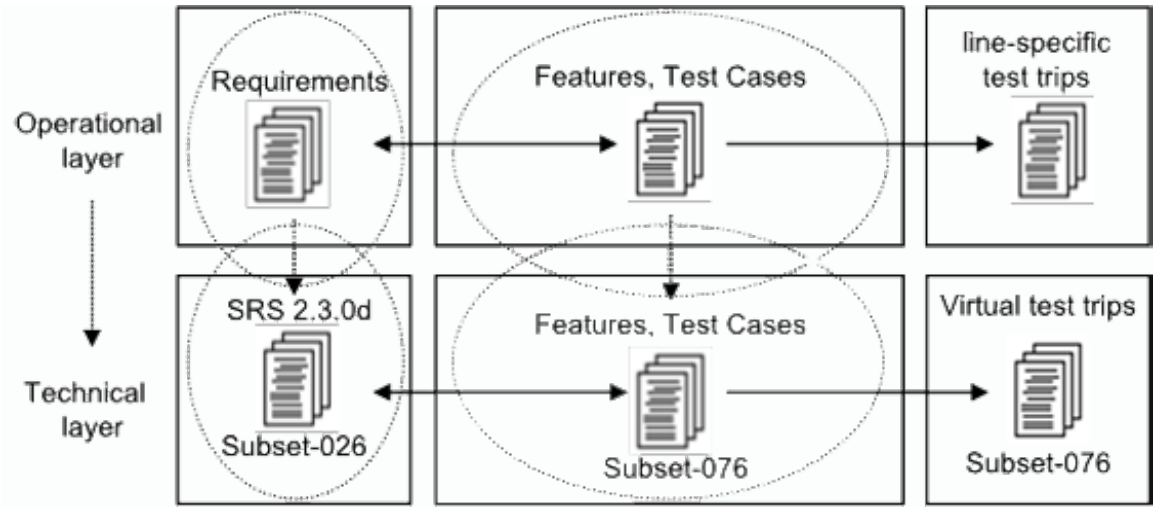
The test execution will take place in one of the three independent Subset 076 test laboratories. One of them is located in Germany at the German Aerospace Centre (DLR) in Brunswick. The two other ones are the labs of CEDEX in Spain and MULTITEL in Belgium.

The DLR Institute of Transportation Systems runs a rail laboratory for the research, testing and simulation of Train Traffic Control and Safety Systems, called RailSiTe® (Figure 7). The lab itself is a complex distributed system. It consists of more than ten computers, running up to 25 distinct software modules. Each module plays a part in a comprehensive simulation of the Railway system on train-side, trackside or air gap.

Figure 7 shows photos of the RailSiTe® laboratory from the DLR. The picture on the left side shows the RailSiTe® Mockup, called RailSET®, which is a complete replica of a driver cabin. The canvas (top left) shows the visualisation of the track (Screenshot at the bottom). Special visualisation software is connected with the lab core to represent the operational part as realistic as possible. The control centre of the laboratory is shown in the right picture.

Figure 8 is divided into three rows. The first row (bottom up) describes the bases of the simulation software. The TSS module supervises the start and stop of all other modules. The Test Scenario Recorder (TSR), is logging all data during the simulation and is the fundament of the test evaluation. The Route Map Controller (RMC) module hosts and distributes all information

Figure 8. Lab infrastructure



about the infrastructure (basically all messages from Track to Train, like Balises, Loops, Radio Messages, Odometrie, Train Interface, etc.). Of course this depends on the SUT. The OBU is used as an example here.

The second row shows the interface layer of the lab architecture. Those modules provide the physical Interface between the lab and the SUT. They can be adapted according the needs.

Communication between the modules has been designed to enable easy integration of several hardware components. Real train- and trackside hardware components can substitute equivalent software simulation modules and thus be integrated into the lab to perform functional hardware-in-the-loop tests.

Therefore data within the lab is transferred over heterogeneous interfaces, which are required to connect to these components, including:

- High Frequency Balise Signal Generator
- High Frequency Loop Signal Generator
- GSM-R Modem Simulation
- ISDN telephone switchboard or Capi as RBC interface
- Serial communication (RS232 / RS485) to generate Odometrie signals on industrial interfaces
- Digital I/O and other physical signals to give feedback from the loco to the Onboard Unit.

The third layer describes the systems, which can be tested as hard- or software. All Subsystems of the Train Control System can either be connected as a Software-Module using Ethernet Interface, as a Software-Module using the physical Interface or as a real hardware.

To cope with the complexity of this system but allow a maximum in flexibility, the architecture has been designed that only logical interfaces are used in the RailSiTe-core whereas the train hardware interfaces, like Loop-, Balise or Radio transmission modules are integrated by a separate hardware adaption layer. The functional correctness of the test laboratory is of great importance for the test of hardware components to ensure their interoperability between different manufacturers. Naturally, a thoroughly tested rail laboratory is the key for these hardware tests.

Using the Subset 076 Test Laboratory to Ensure Operational Interoperability

As mentioned above all technical constraints for executing operational interoperability tests are fulfilled. Finally the root of all specification, the line operation, needs to be represented by an extending set of test cases.

Further it is highly recommended that the methodology of testing of the three levels (i.e. Component, Integration and System Tests) should be compliant and consistent. This means that a similar approach according to the technical test specification is reasonable and should be applied for operational test. In detail at every level there are requirements to be grouped into features which cover all mandatory functional requirements. On the basis of the list of features certain test cases will be created. Hence, you can apply the test cases using proper interfaces to the system under test. The test cases of one level integrate and extent the test cases of the level below. Between the three levels a relationship with a different level of detail is obtained. Thus, it is granted that the test cases will be consistent and at the same time minimizing the testing effort as well as maximizing the benefit avoiding overlapping and repeated test contents.

Beside the defined technology in the SRS, operational tests also define the interaction between technology, operational actors (e.g., dispatcher, interlocking or train driver) and operational rules (guidelines and regulations) as extension to the technical test (Subset-076).

The technical test cases of Subset-076 concern about the functions of the train side (especially OBU) but even though technical functions of the trackside are defined, there are also operational procedures needed and tested.

Figure 9. Testing the top layer: operational requirements

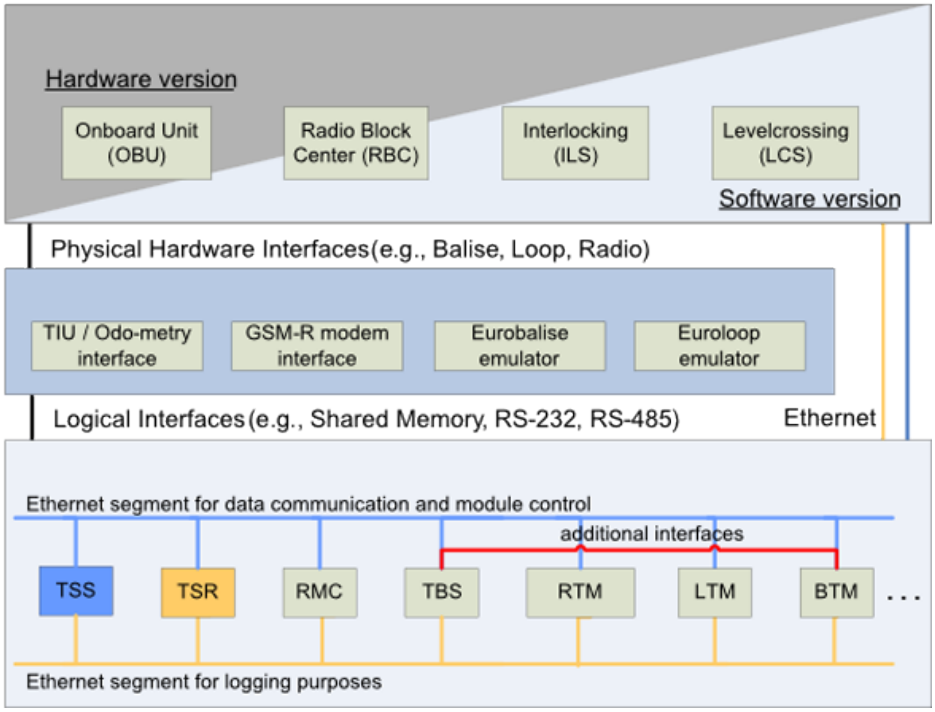


Figure 9 illustrates the context between operational and technical documents. The operational level is the base for the technical level. Test cases of the technical layer can be extended by requirements of the operational layer. Thus it is possible to create virtual test trips not only for the technical but the operational requirements. I.e. any line topology can be tested as a virtual, line-specific, test trip in the test lab. Furthermore the complete set of real RBC and real OBU can already be tested on the virtual implementation of the line in the lab. The Subset-076 test-bench already supports all necessary interfaces, as mentioned before.

FUTURE RESEARCH DIRECTIONS

The next step is to support introducing ETCS in Europe and prevent facing the problems of incompatibilities by making it possible to run complete line tests in a laboratory. Not only OBU Tests will be executed, but RBC- tests as well. Because installing a RBC (the hardware part) in a lab is relatively challenging it will also be possible to leave the RBC at its location next to the track and forward its interfaces into the lab. Furthermore the complete projection of the track will be load into the lab software and a test with real OBU (in the lab) against a real layout (virtual in lab) and real RBC (on the track) can be executed.

This way all technical and operational issues can be minimized and the tracks "down time" can be reduced as much as possible.

CONCLUSION

This chapter has shown the need for operational interoperability to use ETCS across Europe with different Railway Undertakings along the line. Necessarily this leads to additional operational tests to keep the effort as low possible. A general method was demonstrated, how an existing test process can be adapted for the operational requirements.

REFERENCES

Stanley, P. (2011). *ETCS for Engineers*. Hamburg, Germany: EurailPress.

UNIFE. (2011) *ERTMS projects. Website*. Retrieved August 05, 2011from: http://www.ertms.com/2007v2/projects.html.

UNISIG. (2008a). ERTMS/ETCS - Class 1: Test Specification. *Subset*, *076*, 2008.

UNISIG (2008b). *ERTMS/ETCS - Baseline 3: System Requirements Specification*: Subset-026: Version 3.0.0, 23.12.2008.

UNISIG.(2009) *ERTMS/ETCS - Class 1: Functional Requirements for an on board Reference Test Facility*: Subset-094: Version 2.0.2, 05.02.2009.

ADDITIONAL READING

di Tommaso, P., Flammini, F., Lazzaro, A., Pellecchia, R., & Sanseviero, A. (2005). The Simulation of Anomalies in the Functional Testing of the ERTMS/ETCS Trackside System. In: *Ninth IEEE International Symposium on High-Assurance Systems Engineering (HASE'05),* Heidelberg, Germany: IEEE

Ebrecht, L., Meyer zu Hörste, M., & Lemmer, K. (2007). The Basic Concept fort he formal Description – Horizontal Composition and vertical Differentiation of the atomic Element. In Schnieder, E., & Tarnai, G. (Eds.), *Formal Methods for Automation and Safety in Railway and Automotive Systems* (pp. 448–457). Braunschweig, Germany: GZVB.

Fiedler, R., & Didrich, K. (2006). Efficient test method for ETCS components. *RTR*, *2*(February), 2006.

Gralla, C. (2009). *Zur Gestaltung einer ETCS-Migration eines Eisenbahnverkehrsunternehmens* [To make an ETCS migration of a railway company]. Dissertation, Technical University of Braunschweig, Braunschweig, Germany (In German)

Iglesias, J. (2011). ERTMS Successful Deployment in Spain. In *WCRR*. Paris: UIC.

Meyer zu Hörste, M. (2004). *Methodische Analyse und generische Modellierung von Eisenbahnleit- und –sicherungssystemen* [Methodological analysis and generic modeling of railway control and security systems]. Düsseldorf, Germany: VDI. (In German)

Meyer zu Hörste, M., Jaschke, K., & Lemmer, K. (2003). A Test Facility for ERTMS/ETCS Conformity. In G. Tarnai & E. Schnieder (Eds.), *Formal Methods for Railway Operations and Control Systems.* (pp. 281-286). Budapest, Hungary: L'Haramattan.

Theeg, G., & Vlasenko, S. (2009). *Railway Signalling & Interlocking*. Hamburg, Germany: EurailPress.

Winter, P. (2003). *Implementing the European Train Control System ETCS - Opportunities for European Rail Corridors*. Paris, France: UIC.

Winter, P. (2009). *Compendium on ERTMS*. Hamburg, Germany: EurailPress.

KEY TERMS AND DEFINITIONS

Conformity: A system has been implemented according to the European system requirement specification

Corridor: ETCS Line, crossing several countries

Independent Test Lab: One of the three test laboratories in Europe.

Interoperability: Systems made by different manufacturers are working together precisely.

Line: Part of a Railway Corridor.

Operational Interoperability: The two subsystems of ETCS fulfill together the operational functionality

Physical Interface: Same interface as used in the field (e.g., an High Frequency Radio-Communication is emulated in the lab, to send Balise messages via the onboard antenna).

Serviceability: the two subsystems of ETCS fulfil the operational need of the railway

Technical Interoperability: The two subsystems of ETCS interact technically in the correct way

Validation: Proof of the operational usability

Verification: Proof of the correctness and consistency between two development steps

Chapter 6
Fault Injection for On-Board ERTMS/ETCS Safety Assessment

Almir Villaro Arriola
CEIT and Tecnun (University of Navarra), Spain

Jon Mendizabal Samper
CEIT and Tecnun (University of Navarra), Spain

Juan Meléndez Lagunilla
CEIT and Tecnun (University of Navarra), Spain

ABSTRACT

On-Board ERTMS/ETCS equipment performs safety related functions where the tolerable hazard rate is kept below 10^{-9} f/h. Safety standards such as EN50129 or IEC61508 impose requirements on the architecture used to fulfill this safety figure and the associated Safety Integrity Level (SIL). From these standards, the mandatory use of redundancy and physical independence can be derived. Due to the introduction of these requirements, a new functionality is added at the system level (e.g. majority voting processes among redundant lines). Unfortunately, neither the safety nor the interoperability standards provide technical specification that defines how to test the performance of the complete system when internal malfunction has occurred in safety related components. This chapter proposes the use of fault injection techniques to facilitate safety assessment. By means of communication saboteurs, it is possible to excite and test the associated internal functionality in systems performing safety related functions. The chapter also contributes to the definition of the test setup and test procedure of the architecture-associated safety-related internal functionality of the SIL4 odometer and Balise Transmission Module (BTM) subsystems within the on-board European Railway Traffic Management System/ European Train Control System (ERTMS/ETCS).

DOI: 10.4018/978-1-4666-1643-1.ch006

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

This chapter has two objectives; to propose a fault injection technique based on communication saboteurs for carrying out the safety assessment of systems composed of redundant components performing safety related functions, and to contribute to the definition of the test setup and test procedure of the SIL4 odometer and balise transmission module subsystems within the on-board ERTMS/ETCS (Dhahbi, 2011) in order to facilitate safety assessment.

In order to achieve these objectives, the chapter is divided in five parts:

- The characteristics of the on-board ERTMS/ETCS that affect validation are initially discussed. Here the most important requirements derived from safety standards (CENELEC EN 50128, EN 50129, EN 50159; IEC 61508), the building of the equipment and the on-board ERTMS/ETCS architecture requirements are discussed and their implications for the validation of the ERTMS/ETCS are explained.
- The tests needed for the functionality assessment are analyzed, focusing on the on-board ERTMS/ETCS tests.
- The fault injection technique for safety assessment is then discussed. In this part, the characteristics of the DUT with safety related functions are briefly analyzed and the different methods to inject faults in the design are compared. By means of an Failure Mode and Effect Analysis (FMEA) of the system, the effects of the different types of faults are identified. This tool demonstrates that the effect of any fault at component level can be emulated by means of a communication error (corruption, deletion, masquerade, delay, etc.). Moreover, a practical method to inject fault, which enables the validation of systems composed of redundant components performing safety-related functions, is described.
- The fourth part of the chapter discusses architecture-associated safety related to the internal functionality of the odometer and BTM (Balise Transmission Module) subsystems within the on-board ERTMS/ETCS. Currently, the interoperability standards dealing with testing define the black box testing of the ETCS functionality. The authors propose the test setup and test procedure based on fault injection for two SIL4 on-board ERTMS/ETCS subsystems to excite the functionality when a component malfunction occurs. By means of this technique it is possible to facilitate the safety assessment of the on-board ERTMS/ETCS.
- Finally, the last part summarizes the most relevant points covered in the chapter.

CHARACTERISTICS OF THE ON-BOARD ERTMS/ETCS THAT IMPACT VALIDATION

The on-board ERTMS/ETCS is in charge of train control. This system is comprised of a set of functions that enable train command and control such as receiving information from wayside signaling systems (e.g. BTM, Loop Transmission Module or GSM-R), identifying the position and defining the speed profile.

Some of these functions are safety-related, and therefore the involved subsystems shall comply with a set of standards during their complete life cycle. Some of these standards define specific requirements for the on-board ERTMS/ETCS constituents that deal with electrical performance (e.g. EN50155), electromagnetic compatibility (e.g. EN50121 series) or environmental conditions (e.g. EN61373 or EN50125).

Another set of documents (Technical Specifications for Interoperability or TSI) defines the requirements for the different ERTMS/ETCS functions, leaving decisions about the architecture and

the implementation details to the manufacturer. For example, the Form Fit Functional Interface (FFFIS) for Eurobalise (UNISIG, 2007a), defines the Balise Transmission Module functionality for the track-side and on-board parts. It defines in detail the air gap interface and provides quantitative safety requirements for both parts. However, it neither provides a specific interface nor an architecture scheme for the interaction between the on-board BTM with the ERTMS/ETCS Kernel.

Finally, there is a set of standards that, on the one hand, defines the methodology that shall be followed during the complete life cycle of the constituents. This includes all the phases that range from concept and system definition to maintenance or withdrawal. EN50126, EN50128 and EN50129 define some roles and activities for these phases.

On the other hand, standards EN50128 and EN50129 establish some requirements for the system architecture and for specific parts of the design. For example, in terms of architecture, depending on the SIL level the employment of redundant processing channels is mandatory. Other specific techniques are also indicated, such as the use of temperature and program sequence supervision. In addition, there is another standard (EN50159) that defines the requirements for the transmission layers of safety related functions and which makes use of a non-trusted channel. This standard also provides specific technical requirements that the system must comply with.

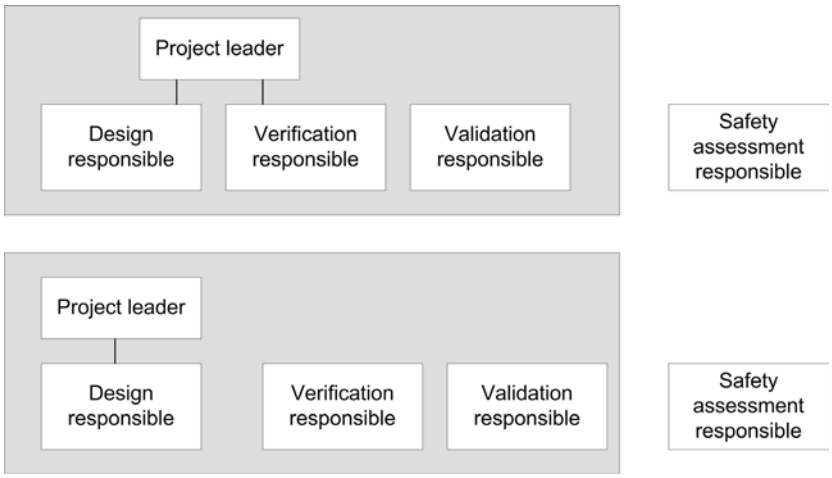
Hence, the validation procedure is greatly influenced by the characteristics of the ERTMS/ETCS. Not only do the characteristics (e.g. SIL level of the functions) define the person responsible for validation within the organization, but also the types of tests to carry out and the architecture or modules to test (e.g. black box testing).

To better illustrate the concepts developed above we can analyze the case of the BTM function within the on-board ERTMS/ETCS functionality. The BTM function is part of the transmission functions within the ERTMS/ETCS. The UNISIG group performed the Preliminary Hazard Analysis and the initial apportionment for the safety objectives (UNISIG, 2009b).

The above analysis assigned the quantitative safety objective BTM-H4 to the on-board BTM function. BTM-H4 is the possible hazard that may occur when a message transmitted from the way-side balises is corrupted and transmitted to the ERTMS/ETCS kernel as valid. The safety requirement indicates that the tolerable hazard rate is lower than 10^{-9} messages corruption per hour. This has the following implications (among others) regarding validation:

- This quantitative objective implies an SIL4 development procedure for the constituents that may present failure modes, which leads to corruption.
- The person responsible for validation shall be independent of all other development roles (design and verification).
- At the system level, the tests to be performed are the most stringent ones, which include a check list, a functional test under nominal conditions, a functional test under the complete range of environmental conditions, and stress testing.
- For the software, the test to be carried out includes functional and black box testing, performance testing and probabilistic testing.
- Regarding the architecture, SIL4 means choosing from among three possible architectures, which involves either redundancy with functional independence or independent additional hardware, to perform supervision of the safety related functions.
- Regarding the detailed design, SIL4 involves the integration of the following techniques in the constituents: protection against functional errors, protection against sabotage, protection against individual faults of discrete components, detection of individual faults, and retention of the safe state.

Figure 1. Organizational possibilities for a SIL3 or SIL4 development process



The on-board BTM function, like other functionalities within the ERTMS/ETCS, may be implemented by employing several devices that communicate internally. The characteristics of these devices generally depend on the business strategy of the manufacturer and therefore, their functionality and internal communications are not standardized. Moreover, the safety function may be split among several devices or the safe reaction for the malfunction may also rely on different devices. The validation testing shall include, for the safety related functions, all the equipment that involve the functionally.

The points presented above are described in detail in the following sections.

Requirements from EN50128 and EN50129

Standards EN50128 and EN50129 define requirements that affect the validation in three different ways: the structure of the organization where validation takes place, the architecture and detailed design that is validated, and finally the test to be performed during validation. These concepts are further described in the following paragraph.

As mentioned previously, the on-board ERTMS/ETCS is in charge of the control of the train, and thus the execution of safety-related functions. Therefore, the life cycle of the ERMTS/ETCS is regulated by EN50128 and the EN50129. These standards define a set of requirements for railway control software and protection systems, and for safety-related electronic systems. EN50129 has a wider scope since it defines the safety requirements of the complete electronic system while EN50128 is focused only on software.

Focusing on the development of a system with safety-related functions, both EN50129 and EN50128 determine a organizational structure with four clear roles, which are presented in the following points:

- The designer in charge of the design of the system.
- The verification team responsible for the outcomes of the different phases.
- The validation team responsible for the validation.
- The Notified Body that performs an independent assessment of the safety case or the evidence obtained during the development process to ensure safety requirements have been met.

The different roles and constraints on the organization in the case of the SIL4 development process are illustrated in Figure 1.

The validation activities do not depend hierarchically on the project leader for devices that execute SIL3 and SIL4 functions. Within this context, generic activities are defined in EN50128 and EN50129 for the roles mentioned above. For example, in the design phase, the designer is expected to perform FMECA, FTA or environmental studies, while in the validation phase the simulation, checking lists, and different types of testing are required depending on the SIL level.

With regards to the architecture and the detailed design, EN50129 states in the section 5.4 that the effect of failures must be demonstrated. The system is designed in such a way that it continues fulfilling its safety requirements in the case of random hardware failure. Standard EN50129 instructs that individual and multiple failure effects be considered during the design. It also defines three possibilities for coping with the effect of failures and meeting the safety requirements:

- **Composite Fail-Safety**: This option is based on at least two components executing the safety-related functions. These are independent of each other, so any common cause failure mode is avoided.
- **Reactive Fail-Safety:** In this option an individual element can perform the safety-related function. The correct performance of this element is ensured by rapid detection and reaction in the case of any dangerous failure. The detection and reaction is independent of the element that has performed the safety-related function. The reaction is undertaken by a second element, and their independence is assured to avoid common cause failure modes.
- **Inherent Fail-Safety:** In this option, the individual element that performs the safety-related function does not present any failure mode that may indicate a hazard.

More specifically, the EN50129 standard defines the architecture possibilities regarding the SIL. Table 1, extracted from EN50129, specifies the architecture requirements.

Table 1 provides different architecture possibilities and methods to achieve the required safety integrity level (SIL). “HR” stands for Highly Recommended, meaning that if that technique is not used, it needs to be justified; “R” indicates that the technique is recommended, and “-” means that the technique does not have to be used. Shaded boxes means that the techniques are alternatives and that at least one must be used.

EN50129 also defines a set of techniques for the detailed design, depending on the SIL. These

Table 1. System/Subsystem/Equipment Architecture from EN 50129

Safety Integrity Level	SIL1	SIL2	SIL3	SIL4
Independence between safety-related systems and non-safety related systems	R	R	HR	HR
Simple electronic structure with automatic tests and supervision	R	R	-	-
Double electronic structure	R	R	-	-
Double electronic structure based on composite fail-safety with comparison of intrinsic safety	R	R	HR	HR
Simple electronic structure based on inherent fail-safety	R	R	HR	HR
Simple electronic structure based on reactive fail-safety	R	R	HR	HR
Different electronic structure with comparison to intrinsic safety	R	R	HR	HR
Architecture justification by a quantitative analysis of hardware reliability	HR	HR	HR	HR

are described in Table 1 of EN50129, and some of them are listed below:

- Protection against operating errors.
- Protection against sabotage.
- Protection against single fault for discrete components.
- Protection against single fault for integrated digital circuits.
- Physical independence within the safety-related architecture.
- Detection of single faults.
- Retention of the safe state.
- Dynamic fault detection.
- Program sequence monitoring.

EN50129, through its safety requirements, introduces new internal functionality to the on-board ERTMS/ETCS that is excited when faults occur and they are verified and validated.

EN50129 and EN50128 define a set of measures and techniques to employ during the verification and validation of the developed system. These measures/techniques are also dependent on the safety integrity level. EN50128 indicates the following ones:

- Probabilistic testing, which is highly recommended for SIL3 and SIL4.
- Performance testing, which is mandatory for SIL3 and SIL4.
- Functional and black-box testing, which is also mandatory for SIL3 and SIL4.
- Modeling, which is recommended for all safety integrity levels.

The techniques and measures indicated in EN50129 have a wider scope because they also involve documentation and more stringent testing. The most relevant ones are:

- Check lists.
- Simulation.
- Functional testing.
- Functional testing under environmental conditions.
- Surge immunity testing.
- Inspection of documentation.

EN50159 Requirements

The ERTMS/ETCS is composed of different subsystems that interact among themselves. On the one hand, in most cases the functionality is split among different devices/tools and on the other hand, safety-related functions require architectures based on reactive or composite fail-safety. This means that they exchange safety-related information. Thus, this exchange has to be done in a way such that the tolerable hazard rate is below the specified value. For this reason, CENELEC (European Committee for Electrotechnical Standardization) issued standards EN50159-1 and EN50129-2. These standards contain specific requirements for communication between two safety-related devices. The difference between them is that the first one deals with a closed transmission system, and the second one with an open transmission system.

A closed system implies the following conditions for the system:

- Only authorized access is allowed.
- There is a known maximum number of connectable users.
- The transmission method is fixed and known.

The key point introduced by this standard is that safety-related systems and non safety-related systems can exchange information employing a transmission system where information can be modified. EN50159 divides the transmission system in two parts; the trusted part and the non-trusted part. In the former, the modification of information is considered possible and usually associated with electromagnetic interferences; in the latter, the modification of information is only considered possible when there is a hardware fail-

ure. In order to identify whether the information has been modified or not, two techniques are used:

- The use of a specific communication protocol in the non-trusted part, which employs messages that contains not only raw data but also some extra information. The extra information is included to make the detection of information modification by the safety-related system feasible.
- Checking the properties of the message in the trusted part of the safety-related system.

The safety-related system can receive information about two kinds of faults: erroneous information and temporal faults. These two fault types can produce a potential error in the safety-related functions. Erroneous information can be due to the type of message, the value (or raw data) in the message or the identification of the sender. A temporal error implies a failure in the sequence of the messages or an excessive delay.

Taking into account the abovementioned possible errors, the following protection measures are included in the architecture of the safety system to avoid risks and detect errors:

- Detection of the sender identification error.
- Detection of the data writing error.
- Detection of the data value error.
- Detection of data not received in a timely manner.
- Detection of loss of communication after a defined time.
- Assurance of functional independence between safety-related transmission function and the layers used in the non safety-related transmission system.

In the communication between safety-related subsystems the authenticity, the integrity and the correct time of the data is ensured. The communication protocol comprises three layers: the physical layer, the link layer and the safety layer. The physical and link layers are considered to be a non-trusted part, while the safety layer is considered a trusted part. In order to identify whether the error types defined just above have occurred, checks are performed in the link and safety layers. Therefore, the following requirements are fulfilled:

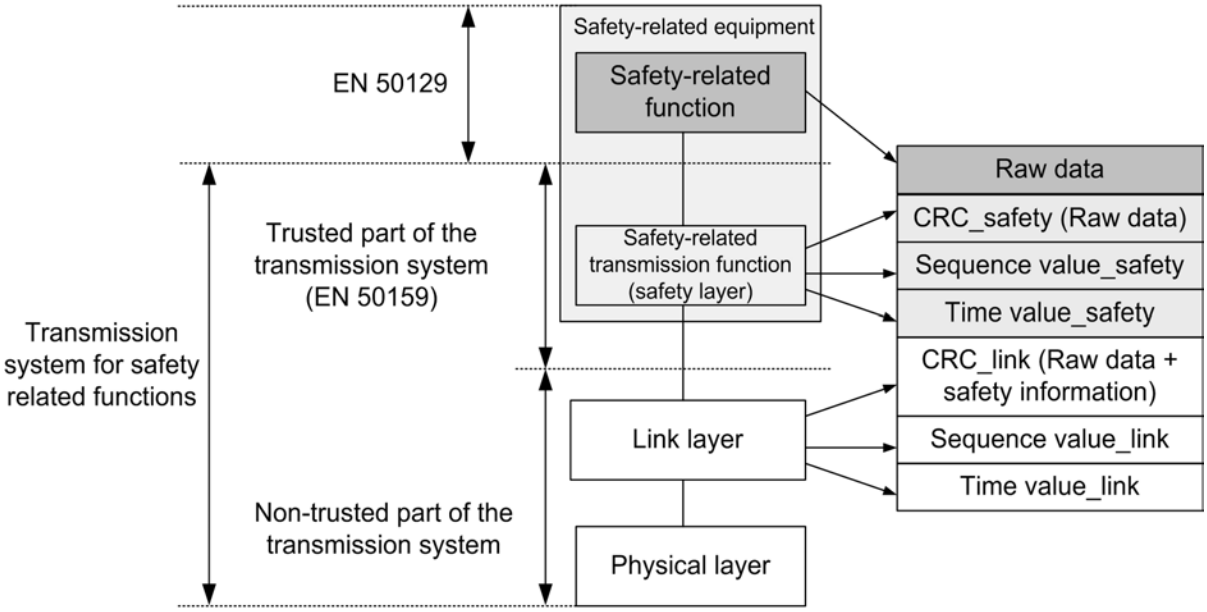
- If the sender is not singularly identified in the transmission system, authenticity is provided by adding the source identification to the user data.
- Integrity is ensured by adding a safety code to the user data. The safety code is not based on the transmission code used, and it is verified by the integrated circuits that are part of the non safety transmission system or link layer.
- The update of the user data is ensured using the temporal information in the user data. The allowed delay depends on the application.
- The safety layer verifies the sequence of the messages.
- The safety layer and link layer employ independent detection mechanisms. If they use the same mechanism different parameters should be used.
- All the previous parameters are controlled, and if the transmission quality falls below a predetermined level the appropriate safety reaction is triggered.

Figure 2 shows the division between the trusted part and the non trusted part of a transmission system where safety-related functions are involved. Moreover, it proposes some specific techniques to deal with the safety requirements introduced by EN50159-1, extracted from (Idirin, 2011).

IEC EN61508 Architectural Requirements

It is worth mentioning that there is an international standard which defines the safety requirements for

Figure 2. Trusted part and non-trusted part of a transmission system



electrical, electronic and programmable electronic safety-related systems.

IEC EN61508 is the general standard for functional safety, and it is applicable to all types of industry. The specific standard for a particular environment can be more restrictive, but should not disagree with it.

Like EN50129, IEC EN61508 defines a set of requirements that affect the validation of the safety-related systems in three ways: (a) it defines the person responsible for validation for the organization of the development project, (b) it defines new functionalities to be validated through the techniques employed in the architectures of the system and detailed design, and (c) it defines a set of tests to be performed during the validation.

However, the process to set the requirements in IEC61508 presents some differences for EN50129. IEC 61508 introduces the requirements in terms of the type of system, the safe failure fraction (SFF) and the SIL.

- Part 7.4.3.1 of the standard IEC EN61508 defines the safety system classification as type A or type B. Type A systems fulfill the following requirements:
  - The failure modes of all the constituent components are well defined.
  - The behavior of the system under fault conditions can be completely determined.
  - There is sufficient data to show that the claimed rates of failure for detected and undetected dangerous failures are met.

If one of the requirements is not met, the system is classified as type B. All the subsystems with computational parts are type B because the failure of processors cannot be completely defined.

- The SSF is the percentage of failures that do not pose any risk to the system because they are detectable or non-dangerous. The SFF can be increased by employing a set of proposed techniques.
- The SIL does only not depend on the tolerable hazard rate but also on the operational

Table 2. Hardware safety integrity: architectural constraints on type B safety-related subsystem, IEC EN61508

Safe Failure Fraction	Hardware Fault Tolerance		
	0	**1**	**2**
<60%	Not Allowed	SIL1	SIL2
60% - <90%	SIL1	SIL2	SIL3
90% - <99%	SIL2	SIL3	SIL4
>99%	SIL3	SIL4	SIL4

characteristics of the equipment (high demand and low demand).

Table 2 shows the type B safety systems with the corresponding architectural requirements.

Hardware fault tolerance means that the safety function execution continues in the presence of faults. Hardware Fault Tolerance 1 means that the system cannot lose the safety functions in the presence of one fault. This implies an architectural design of at least one processing line with diagnostic and safe reaction or two voted processing lines with safe reaction. A SIL4 system always uses hardware fault tolerance and has an SFF higher than 90%.

Regarding the validation responsibility in the project's organizational structure or the requirements for the architecture of detailed design, IEC61508 and EN50129 are quite similar, although the testing requirements in the validation are better defined by IEC61508. For example, in the fault injection testing, Table 2 in part 2 of IEC61508 indicates that fault insertion testing is considered highly recommended for all SIL levels when the required diagnostic coverage is above 90%, where diagnostic coverage is the fraction of dangerous failures detected by automatic on-line diagnostic tests.

In addition, EN50129 states that functional testing is required for validation. It specifically indicates that a complete functional test, based on a well-defined test, should be carried out to demonstrate that the specified characteristics and the safety requirements are fulfilled.

Building Requirements

The railway industry has specific requirements, such as modular design, industry driven standards, maintainability or cost saving, which have to be combined with the safety requirements.

The systems have a modular design so any part with a malfunction can be quickly substituted, while the correct ones are kept.

The use of widespread standards (for interfaces, assembling, manufacturing, etc) is also recommended from the safety point of view, as products meeting these standards are usually well proven in use. From the manufacturers' point of view, it is also beneficial since their products can be compatible with other systems. Moreover, the use of standards increases the number of possible suppliers, thus decreasing the cost of the components.

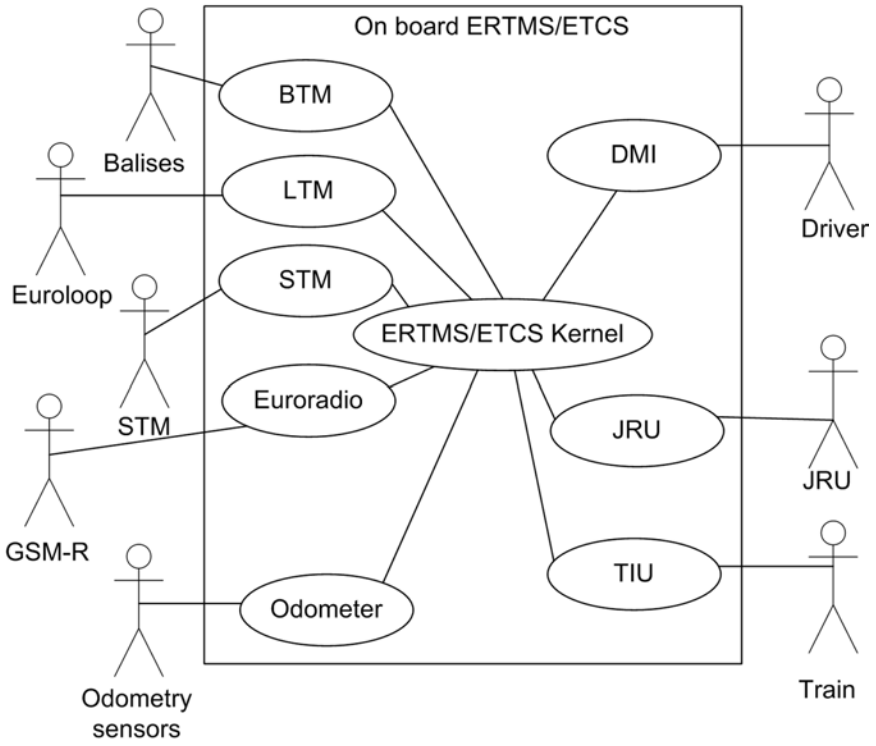
Therefore, one common practice in safety-related equipment is to employ a set of replicated boards with standardized dimensions that are interconnected through a passive backplane employing one standardized communication protocol.

On-Board ERMTS/ETCS Architecture Requirements

The on-board ERTMS/ETCS is composed of three types of functions:

- A set of transmission functions such as the BTM (Balise Transmission Module), LTM (Loop Transmission Module),

Figure 3. On-board ERTMS/ETCS functions and actors



STM (Specific Transmission Module) or Euroradio (GSM-R), which provides the ERTMS/ETCS Kernel with information provided by the wayside signaling system.

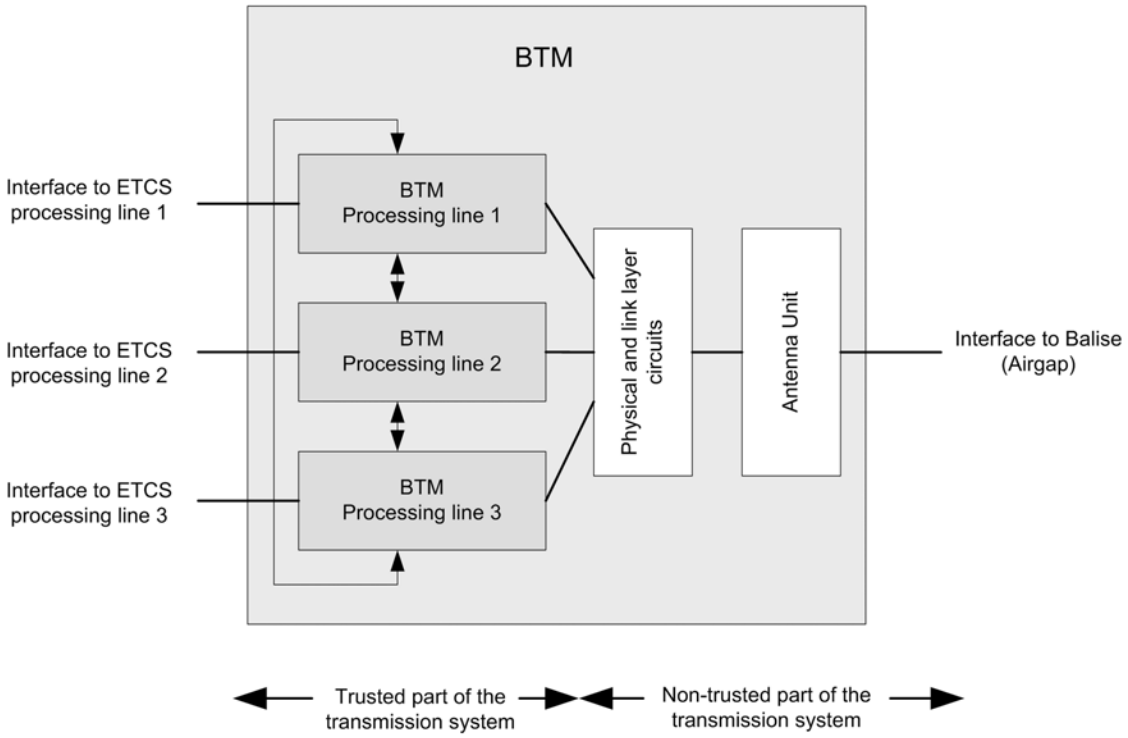
- A set of functions such as the odometer, JRU (Juridical Recording Unit), DMI (Driver Machine Interface) and TIU (Train Interface Unit), which captures and provides relevant information from the ERTMS/ETCS Kernel to the train and vice versa.
- A function called the ERMTS/ETCS kernel, which processes the information provided by the wayside signaling system through the transmission functions and thus commands the train.

Figure 3 represents the different actors and functions in the on-board ERTMS/ETCS context.

The ERTMS/ETCS has the role of providing the driver with the information that allows him to drive the train safely. The core hazard for the ERTMS/ETCS is the "Exceedance of the safe speed/distance as advised to ETCS" and it has an overall safety target (THR_{ETCS}) of $2 \cdot 10^{-9}$ dangerous failures per hour per train (UNISIG, 2009b). To provide information to the driver, several functions cooperate together. Therefore, the safety objective is divided among the on-board functions, the transmission functions and the wayside functions. The hazards of the different functions contributing to the core hazard are identified in the document (UNISIG, 2009b) and a safety target for the functions is derived. Thus, all the functions are assigned a THR and an SIL.

To meet the safety requirements, the manufacturers must design the equipment that executes the functionality according both to standard EN50129 and to some other constraints given by business aspects (e.g. availability, cost or location in the train).

Figure 4. Possible architecture of BTM functionality



For example, the BTM function may display the hazardous event BTM-H4. This hazardous event describes the case where the message transmitted from the wayside balises gets corrupted and transmitted to the ERTMS/ETCS kernel as valid. The safety requirement indicates that the tolerable hazard rate is lower than 10^{-9} corrupted messages per hour. On the other hand, the document EEIG 96S126 sets the unavailability of the BTM function below 10^{-8} per hour.

To meet the functional, manufacturing, safety and unavailability requirements, as well as the ones derived from standards EN50129 and EN50159-1, it is possible to employ the architecture described in Figure 4.

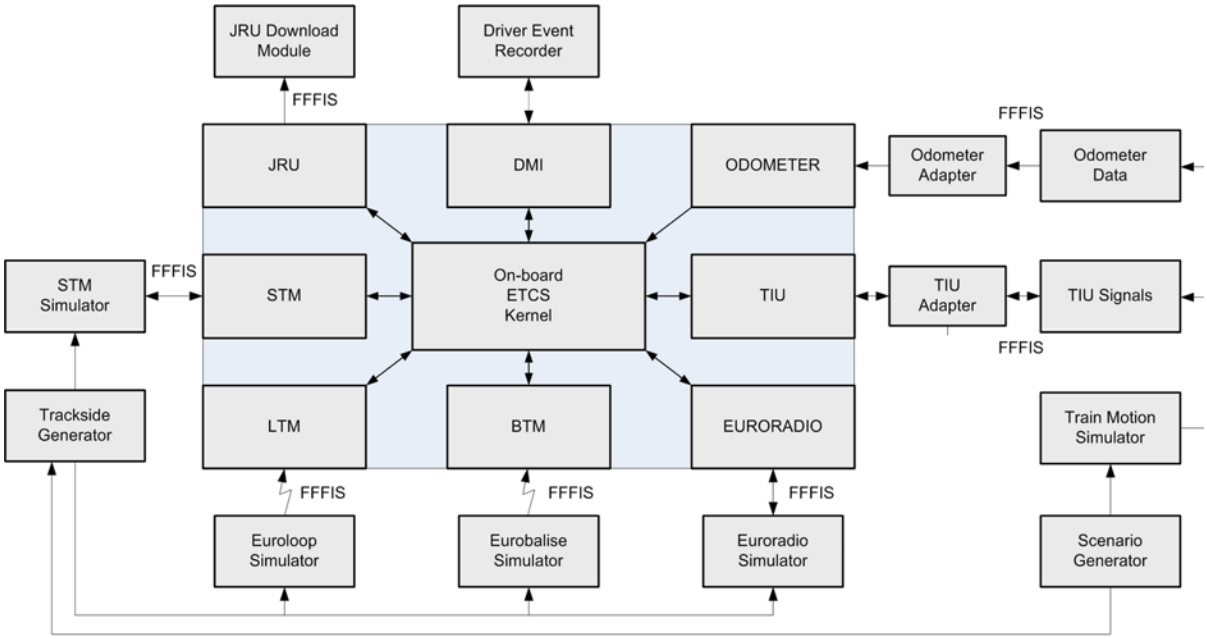
Test Specification Deficiencies for Safety Assessment

The systems for railway safety-related environments are designed against failures that do not occur frequently during normal operation. If tests to the system are only done as a black box (inputs versus outputs), the functionality that implements safe functions cannot be easily evaluated. Test specifications for on-board ERTMS/ETCS, (UNISIG, 2007b, 2009a, 2009c), are not designed to introduce faults in the system. There is no technical specification with clear guidelines about how to test safety even though it is tested, as stated in the previously analyzed standard, such as IEC611508 or EN50129. Other test strategies, apart from the technical specifications, have to be carried out and developed. Internal faults inside the on-board ERTMS/ETCS systems are injected to provide evidence to the Notified Body.

On-Board ERTMS/ETCS Test Environment

The on-board ERTMS/ETCS is composed of various functionalities. It has functionalities for

Figure 5. On-board ERTMS/ETCS equipment test Architecture



reading balises in the trackside (BTM) and obtaining messages from euroloops (LTM), as well as for specific signaling systems (STM). It shows information to the driver and it accepts some commands from him (DMI). It interacts with the train to brake and perform other actions (TIU). It obtains measurements about distance travelled and speed (Odometer). It communicates with the movement authority through the GSM-R standard (Euroradio) and it records the necessary juridical information (JRU).

Some of the interfaces of the on-board ERTMS/ETCS that accomplish such functionalities are standardized; these interfaces are defined in the FFFIS of the different functions. There is an FFFIS for the downloading of Juridical Data from the JRU, for communication with Eurobalises and Euroloops situated in railways, for GSM-R communication, and for communication through the Specific Transmission Module. For other subsystems such as the Train Interface Unit or Odometer there is not a standard; instead they are company specific, but a test interface is defined. Therefore, it is necessary to introduce some adapters between the test environment and the on-board ERTMS/ETCS. This is shown in Figure 5. For the Driver Machine Interface there is no specification, so it should be manually or robotically operated. It can be seen that internally the on-board ERTMS/ETCS has several communications.

(UNISIG, 2009a) specifies the methodology to be used to test the on-Board ETCS. Thus, the fulfillment of the functionality described in (UNISIG, 2010) is assured. An example of test environment architecture is shown in Figure 5. This architecture fulfills the functional requirements of (UNISIG, 2009c) for the test facility in which all the sequences of tests from (UNISIG, 2009a) can be evaluated. The test is commanded by the Scenario Generator, which controls the Train Motion Simulator as well as the Trackside Generator. The Trip Analysis obtains all the relevant information from the system so that the system response, defined as outputs in the test sequences, can be evaluated and compared to the reference response. For each FFFIS there is a simulator. The on-board

ERTMS/ETCS manufacturer is responsible for providing an adapter for the Odometer and for the Train Interface Unit.

With such a system all the tests specified in the standard are black box tests, and the system is treated as an indivisible system. The communications are standardized to obtain information from outside the system. Thus, it appears unfeasible to introduce any fault inside the system and to test safety reactions since it is not defined in the standard. The system can be evaluated with errors that come only from the inputs, such as a badly constructed trackside with inconsistent information. The loss of a balise, simulating a damaged one, or an erroneous information transmission can be evaluated. But a failure in the communication of information due to an internal failure cannot be evaluated, and consequently the possible failure modes of the system cannot be evaluated.

Internal communications between functionalities, e.g., the communication between the BTM and the Kernel are company specific. However, each FFFIS is evaluated in a specific environment that can reproduce all the working situations of the FFFIS interface. The Eurobalise FFFIS are defined in the document named (UNISIG, 2007a) and the testing conditions and environment are defined in Subset 085 (UNISIG, 2007b) looks for testing the performance of the BTM functionality at the output interfaces. Thus, the company specific communications shall be adapted to the testing interfaces. In addition to the on-board ERTMS/ETCS, the BTM functionality is evaluated as a black box and safety requirements that affect only this part are not evaluated.

The entire test norm for the on-board ERTMS/ETCS treats the systems or functionalities as black boxes. The functionality of the system defined in the norm is evaluated, but the extra functionality that implements reactive or composite safety is not evaluated, and as a consequence the fulfillment of EN50129 or IEC61508 is not assured with these tests.

Fault Injection Technique for Safety Assessment

To obtain the validation of the on-board ERTMS/ETCS, all the system functionality (e.g. BTM, LTM, GSM-R, JRU and DMI) shall be evaluated, including the extra functional requirements added to accomplish the safety objectives. With black box testing, it is impossible to evaluate performance of the system in the presence of failures because either we do not have any mechanism to provoke in a control and repeatable way internal faults, nor faults will spontaneously occur due to the extremely low probability. Therefore, it is necessary to introduce faults into the system and evaluate the response.

DUT with Safety Characteristics

A generic safety system is used for the selection of an appropriate method for the validation through fault injection. This generic safety system is a voter with a two-out-of-three scheme, Triple Modular Redundancy (TMR), which provides fault tolerance and safety. It must meet all the requirements presented in section 3. Using three independent processing boards, it provides a fault tolerance of one. One of the boards can produce an erroneous result while the others, for the most part, provide the correct answer. The architecture of the system can be seen in Figure 7. Communications are routed through a passive backplane or by wire, using Cat5 cables or shielded twisted pairs, among others.

The three boards will communicate the information to be voted to each other, so the voter is implemented by software in each board. There is not a unique voter, so common mode failures in the voter are avoided. This provides a safety scheme like the one demonstrated in (Idirin, 2011). Communications between boards shall accomplish the requirements derived from EN 50159-1 to protect the safety related functions from the possible errors introduced in the non-trusted communication

channel. The communication employs three layers: the generic link layer protocol, the EN 50159-1 safety layer and the application layer. In the safety layer, to identify possible sources of errors, a set of additional information is employed such as the sequence number, timing information, CRC safety, and sender identification.

The transmission of information to be voted is usually asynchronous, and thus the transmission media used can be an asynchronous one. It can use standard communications such as RS485 or Ethernet, and it can be rack mounted. The transmission of information is immediate when the event occurs, which means that a protocol for avoiding collision can be used in the link layer. The direction of interchanged information between processing lines can be defined, for example, first clockwise and then counterclockwise.

Methods for Injecting Faults in Design

There are several methods for injecting faults (e.g. heavy ion irradiation, simulation or on-line register modification), and the methodology differs depending on the stage of product development and on the characteristic of the performance to characterize.

For example, fault injection can be used during the design stage of integrated digital circuits when the first prototypes are available, previous to the final hardware implementation. In this situation, to characterize the susceptibility of the future product to Single Event Upsets (SEU) caused by heavy particles, the prototypes can be irradiated with heavy ions or laser beamed (Arlat, 2003; Dutertre 2011). The results can be used to take the decision of enhancing the design with internal TMR techniques (with a higher cost) or not. Unfortunately, this method lacks of controllability and repeatability. During the test it is impossible to ensure the exact location where the heavy ion colliding. In order to mitigate this situation statistical analysis is employed (Arlat, 2011). Nevertheless, when the failure rate to characterize is in the order of 10^{-10} failures/hour, the cost of the statistical analysis is huge due to the special facilities required and the number of prototypes.

In the design stage, faults effect can also be analyzed through simulation. Faults can be injected in a executable model and the performance of the design simulated (Baraza, 2005; Benso, 2007). When the effects of specific faults of the programmable logic are desired to be analyzed, the nominal description in the VHDL language of the design should be modified. In these cases saboteurs or mutations can be used. Mutations are an alteration of the code, where a part of the code is selected and substituted by a faulty one. Saboteurs are introduced into the code and change the information in the transmission of one module to another. When the effects of specific faults of a certain computing unit, where a software design is intended to run, are desired to be analyzed, similar options can be employed (Martins, 2000; Jinfu, 2007; Vinter, 2007). In this case, the functionality is ultimately given by the SW description. Therefore, the saboteurs or the mutations will be introduced in the SW model to emulate the HW failures. Both methods present unique features in terms of controllability and repeatability. Unfortunately they do not employ the final product either the nominal design, or in other words, both are intrusive and only employ a model of the design.

Another possibility to evaluate the performance of a design in the presence of HW faults is the on-line modification of internal registers. A software design running in a computing unit, or a programmable logic design being executed in an FPGA can be evaluated against the tolerance of failures by modifying certain internal registers. In the case of software, the introduction of faults is usually done through the Test Access Port (TAP), which is used to debug applications (Carreira, 1995; Skarin, 2010; Fidalgo, 2006). This port gives access to the inputs and outputs of the processors and to the registers. Unfortunately, the

fault injection is dependent on the configuration of the TAP, and for the most common one, JTAG, it is necessary to stop the execution of the system in order to access and change an internal register. This occurs because TAPs are designed to debug software applications running in the system and it requires stopping the execution and analyzing the registers and memory. The evaluation is dependent on the kind of TAP that is implemented. Newer TAPs allow access registers synchronized with the program execution but it is necessary to record the exact moment of injection so it can be repeatable. The generic voter is composed of various processors, making it difficult to reproduce the exact moment of injection and the state in each processor. Moreover, the range of possible HW faults is limited to the modification of certain registers, for example, most TAPs do not access the program stack, and therefore the performance of the design in the presence of unexpected state transitions could not be evaluated.

In the validation of the ERTMS/ETCS system, tests shall be done with minimal intrusiveness and it shall be justified that they do not have an impact on the characteristics of the system. Otherwise, it will not be appropriate for certification. Tests shall be also reproducible, in case the Notified Body requires evidences of the tests. It can be concluded that previously analyzed techniques are too intrusive for the validation of the ERTMS/ETCS.

An FMEA of the System

In order to introduce faults for safety validation of the system, the first step is to decide which faults are going to be injected into the system. Therefore, it is necessary to make a study of the possible faults that affect the safety function. For the evaluation of any fault and its consequences, an FMEA analysis is carried out, seeing how a fault has consequences at different levels. For the generic voter, the different levels are: component level, function level, processing line level and system level, as the example in Table 3 illustrates.

All the faults are originated in a component on the processing board. Faults can appear in memories, in processors or in non-processing components, such as oscillators and capacitors. The probability of failure for each component shall be used to know if it is dangerous or causes the non-compliance with the safety level objective. Examples of the MTTF, Mean Time to Failure, of components (RIAC) are:

- Hold Up capacitors (Failure rate: $3{,}7x10^{-6}$ f/h).
- Memories (Failure rate: $74x10^{-9}$ f/h).
- Microprocessors (Failure rate: $75x10^{-9}$ f/h).
- FPGA (Failure rate: $3x10^{-9}$ f/h)

After analyzing all possible failures that threaten the safety objective (discarding the parts that do not pose a danger to the safety objective), the effects of the failure of this component are analyzed at the function level. At the function level the consequences of the processing of information is evaluated. Some examples are:

- **Hold Up Capacitors:** Faults in capacitors can produce instability in the microprocessor's power supply and therefore lack of processing or corruption in processing information.
- **Memories:** A fault in the memories can cause an interruption in the execution of the microprocessor, affecting the program flow or the correct processing of erroneous information stored in memories.
- **Microprocessors:** A fault can cause the interruption of processing or the incorrect processing of information.

The processing line level for the system analyzed is when communication with different parts is affected. When the consequences of a fault are

Table 3. Example of Generic FMEA of a TMR system

Effect at system level	**Divergences between processing lines solved by the voting process** **Divergences in one of the processing line outputs**	**Divergences between processing lines solved by the voting process**	**Divergences between processing lines solved by the voting process**	**Divergences between processing lines solved by the voting process**	**Divergences between processing lines solved by the voting process**
Effect at communication level	Deletion of input/output message Corruption of input/output message with incoherent safety or link layer	Delay of message with valid safety and link layer Insertion of incorrect information in a message with valid safety and link layer Erroneous transmission of a message with valid safety and link layer Repetition of message with valid safety and link layer Message modification with incoherent safety or link layer	Deletion of input message Deletion of output message	Corruption of message with incoherent safety and link layer	Corruption of message with coherent safety and link layer
Effect at function level	Real faults and not a model in the system. Faults are injected in Real Hardware and Software. No modification to the system is done	Application information corruption Processing information modification Safety or link layer corruption	No reception of information No transmission of information	Information modification	Information modification
Fault	Electromagnetic Transient affecting communication lines (EFT, Surge, ESD)	SEUs in RAM Based circuit (Memories, Microprocessors, FPGAs)	Pin Stack	Cold Solder Joint in Input/Output	Cold Solder Joint in Memories

propagated to other parts of the system, in this case, to another processing line, the following can occur:

- Interruption of processing: Omission of messages between processing lines.
- Incorrect processing of information: Omission of messages, corruption of protocol, corruption of information between processing lines.

Most of the potentially dangerous failures will be detected by the protocol as specified by the EN50159-1 standard. These corrupted messages will be rejected in the protocol checking process of the processing lines. However, there is still one kind of failure that will not be rejected by the protocol checking but must be detected at the system level: information can be corrupted before the encapsulation in the protocol. Thus, the protocol is correct but the information in the application layer is erroneous, and this erroneous information reaches the other processing lines.

In an SIL4 system, one internally originated fault cannot be propagated to the output of the system. Therefore, the internal erroneous information that is propagated to the other processing element and not filtered by the protocol checking shall be filtered employing the majority voting process, which is the safety functionality that shall be tested.

Method to Inject Faults in Validation

The FMEA shows that modifying the communications among components, the consequences of dangerous faults at low level can be reproduced. Therefore, emulating faults, the response of the complete system in the presence of these faults can be observed. Faults at low level without consequences are not emulated. Fault injection can be carried out with saboteurs in communications, using the same concept as seen in VHDL evaluation between modules (Baraza, 2005). Saboteurs capture messages and then release them with altered information that fulfills the protocol.

The advantage of these saboteurs against the mere corruption of bits directly in the interface is that all the high level fault states can be reproduced. This technique does not reproduce the typically common and inoffensive pin level faults but rather it reproduces the few extremely dangerous ones (Figure 6). The opposite strategy (Blanc, 2009), which consists of exciting all the pin level faults, is a very time-consuming task and only produces few problems at application levels, as the intermediate safety measures will solve most of the problems. Other methods (e. g. simulation or software fault injection) are not appropriate for safety assessment by a Notified Body as they are intrusive in the design.

The fault injection proposed depends on the communication media. Saboteurs for fault injection in communications can be developed for various types of communications, e.g.: serial communications or star topology communications. A saboteur is an element that is placed between two components and thus, it breaks the communication path and changes the information between modules. As shown in Figure 7, the saboteur affects neither the software nor the hardware of the processing lines. One possibility with minimal intrusion in the equipment is to employ the typical passive backplane, which only interconnects redundant processing boards, to introduce the saboteur.

As saboteurs capture information and then have to transmit the modified information, the transmission time is doubled. In most of the cases this does not suppose actually a restriction on using saboteurs. Usually the internal communications of ERTMS/ETCS systems are designed to fulfill the requirements of all the subsystems transmitting simultaneously which barely occurs. Thus, the communication capability is oversized and a safety margin is introduced to avoid collisions. Moreover, the asynchronous or sporadic nature of the external communication makes the design of saboteurs easier.

The faults injected into the system correspond to those analyzed in the FMEA: message omission, protocol corruption (incorrect variables in safety and link layers), message corruption, information corruption (with correct safety and link layers), message duplication, message delay.

The tests are carried out by comparing the results obtained in the fault injection campaign with a nominal behavior of the system. Firstly, to record the nominal behavior, the test is done without any fault injection. After that, to evaluate the internal functions that provide the availability of the system, faults are injected is in one processing line. In this case, the external behavior of the system shall be the same as the nominal one. In this way the fault tolerance can be demonstrated. Finally, the corruption of various lines is done, and in that case, the safe reaction must be seen as the output of the overall system. In most cases a silent state is the designed safe state.

Fault Injection into the On-Board ERTMS/ETCS

The test environment for the on-board ERTMS/ETCS safety assessment is composed of the test architecture presented in section 4 for inputs and outputs analysis and the fault injection method presented in section 5 to evaluate the availability and the safe reaction.

Figure 6. Comparison of reachable level with fault injection

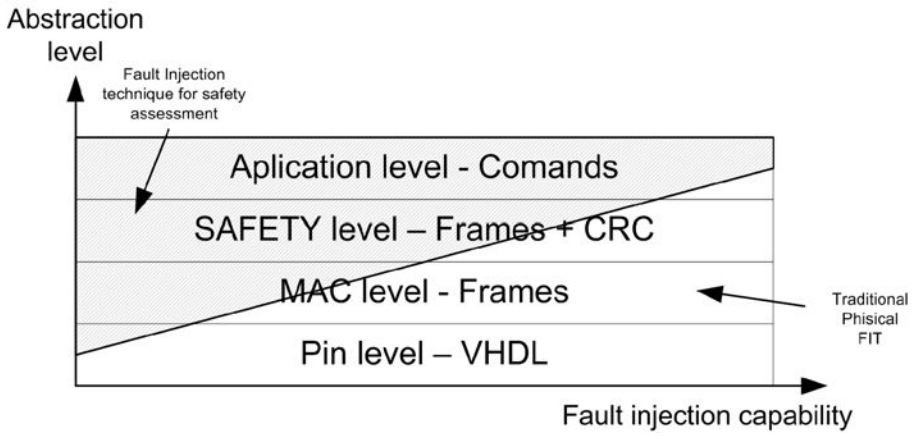
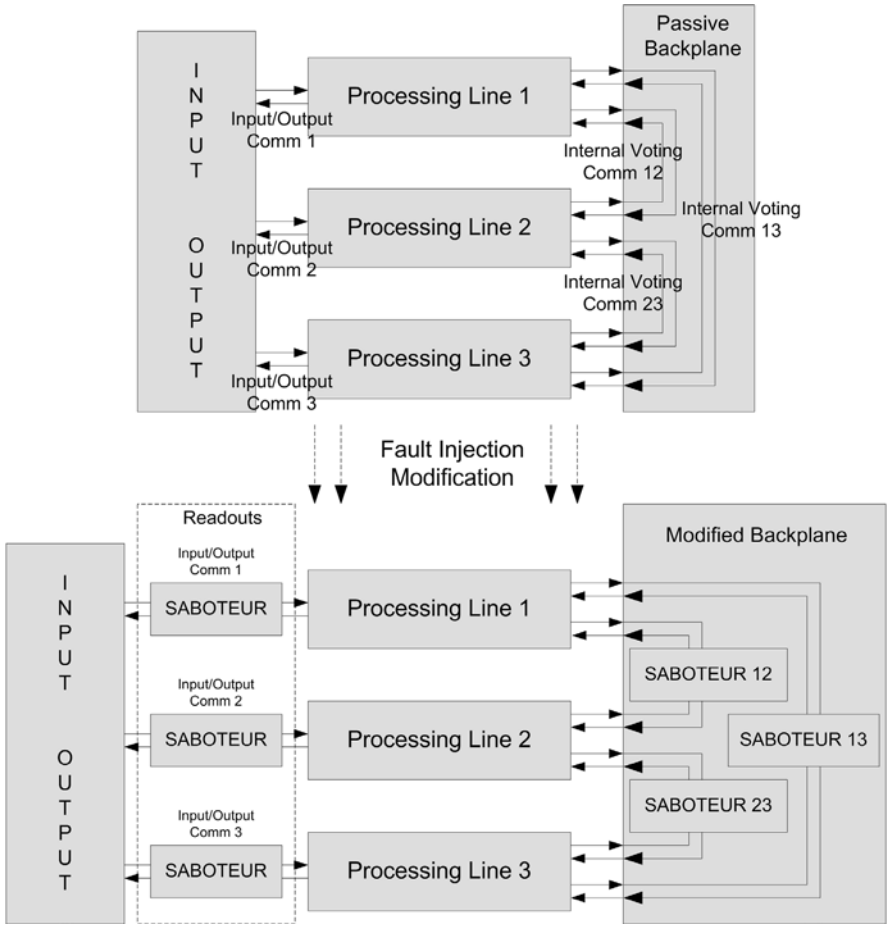


Figure 7. Device under test and modified DUT with Fault injection Tool



ERTMS/ETCS has multiple safety functions, thus the use of voters is commonly employed in the architecture. The information processed by some voters is related to the trackside information. It generally arrives sporadically, so voters can be developed using asynchronous communications. The generic example of the voter inside the on-board ERTMS/ETCS can be used, and the delays of the saboteurs do not affect the system. Moreover, when safety tests are carried out, the whole subsystem does not have to communicate at the maximum information rate.

The designers of the on-board ERTMS/ETCS have many possibilities and should evaluate the different trade-offs among cost, performance, availability and so on. For example, the communications can be implemented using point-to-point transmission. This means that in order to transmit any information between subsystems, a physical path must be implemented. This supposes higher cost that other topologies but removes the collision of messages. Regarding the physical path there are also several options e.g. RS232, RS485, UART communication (Universal Asynchronous Receiver Transmitter). Again a new trade off is posed between bit rate and immunity or bit error rate.

Point-to-point physical transmission can be easily corrupted, as the communication is asynchronous and sporadic. The only information that goes through the sabotage corresponds to the subsystems that need to be attacked/eradicated.

Another implementation possibility is the use of star topology with a switch in communications, although the logic path of communications is not multicast. The communications are point-to-point but using a physical star topology. These communications minimize the number of wires used, although it increases the possibilities of information collision considerably.

If every subsystem (odometer, BTM, DMI, etc.) is communicated through the switch with the kernel, one physical channel processes a lot of information. The correct place to situate a saboteur is not between the switch and the kernel. Instead, saboteurs are situated between the subsystem to corrupt and the switch, meaning the saboteur does not have to process all the information, just that which corresponds to the subsystem. As a consequence, the delay will be minimized.

The communications of the subsystems, such as the BTM or the DMI with the Kernel, shall be known by the validation group because the subsystem shall be also separately validated (section 4.1). The BTM shall fulfil the test for the eurobalise FFFIS, and the correct information communication with the Kernel is evaluated there. The validation group knows the communications because an adapter shall be developed. If there is a communication and its parameters are known, a saboteur for that communication can be used.

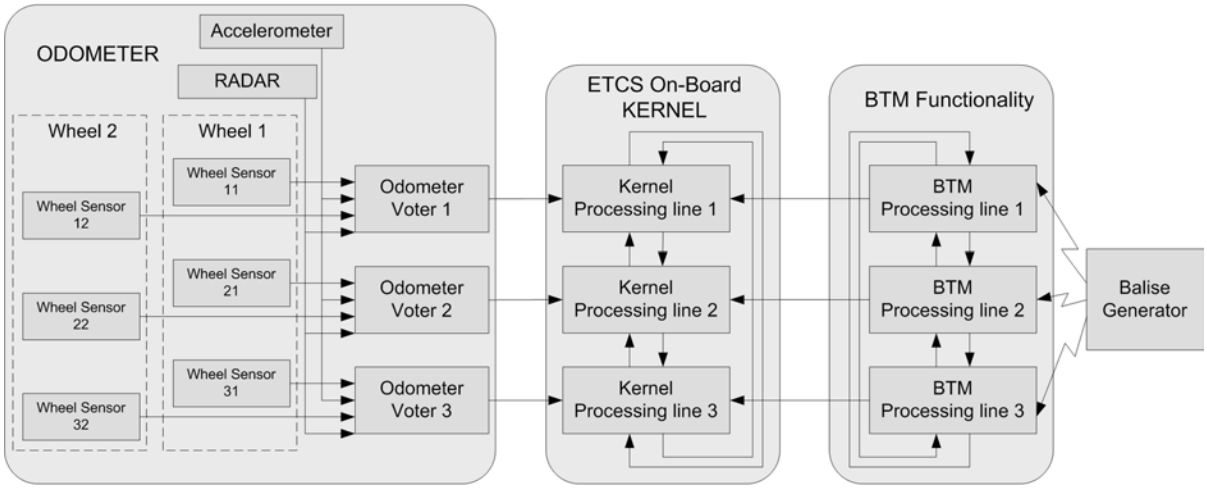
The operation of saboteurs does not depend on the information input or output to the ETCS. It depends on the information that flows from the BTM to the Kernel. Thus it analyses the information and passes it without alteration until the information desired for corruption is detected.

When the information of a balise needs to be corrupted it awaits a message with the balise information needed to do so. It can be programmed to detect a specific message of a balise; thus, the other balises pass the saboteur but the selected one does not. In this way, the availability or the safe reaction of the system can be evaluated without altering any part of the on-board ERTMS/ETCS.

The architecture to be tested can present a distribution in the case of odometer and BTM function similar to the one shown in Figure 8. The odometer is composed of three-out-of-four voters, and it is voted again in the ETCS kernel with a two-out-of-three voter (Stamenkovic, 2009).

The BTM function has safety functions that force it to use at least a two-out-of-two voter. This kind of voter can have an architecture similar to the generic voter presented in (Idirin, 2011). The ETCS kernel can also be constructed with the architecture of the generic voter. Thus, the system

Figure 8. Part of the architecture of on-board ERTMS/ETCS composed of voters to be tested



is performed by elements that can be corrupted with saboteurs.

The tests to obtain safety and availability functional validation can be based on the standard that describes how to test the on-board ERTMS/ETCS. (UNISIG, 2009a) is composed of multiple test sequences that can be adapted. The saboteurs can be programmed to modify the information in the communication lines. In this way, comparing the generated stimuli and the recorded response of the system, with a reference, the functions that provide the safe reaction and those which enable the system availability can be validated.

The injection hardware for the whole system does not present a significant cost, as it can be done with Linux servers or with communications evaluation boards, in contrast with the cost of a balise simulator for the BTM module certification.

Fault Injection in the Odometer

The architecture where faults are introduced is the odometer, the BTM and the Kernel example presented in Figure 8.

In the validation of odometer availability and safety, failures are introduced both at the odometer and ETCS levels. The three-out-of-four voter and the odometer data from the Kernel's two-out-of-three voter is evaluated.

The company should provide an adapter that transforms a standardized signal of speed control as defined in (UNISIG, 2009c), into the signals of odometer radars, accelerometers or wheel sensors installed in the train. This adapter shall be designed to introduce faults into any channel of the three-out-of-four voters. Thus, the odometer level is analyzed with an adapter and not a saboteur.

The odometer at system level is evaluated using saboteurs between each processing line of the kernel. The information exchanged relative to the speed and odometer is attacked.

(UNISIG, Subset 94) specifies test sequences to evaluate the On-Board ERTMS/ETCS system against its functional requirements. In these sequences the inputs of the systems are defined in addition to the outputs. The execution of a test is the execution of the inputs. The validation of the test is the comparison between the outputs of the test execution and the outputs defined in the (UNISIG, 2009a).

Any sequence in the Subset 076 standard uses different kinds of speed. Thus, it is suitable to use any test sequence as a template to evaluate odometer safety. The new sequence consists of the three consecutive executions of a sequence.

In the first run, no fault is applied, in the second a fault is applied in one channel, and in the third execution, two unrelated faults are applied in different channels. At the odometer level the faults are produced by the special adapter, and in case that the test is at the ETCS level, the faults are produced by saboteurs.

When one fault is applied the response shall be similar to the execution of the test without faults. In any case, the functionality shall be unaltered and the outputs shall be the same as the (UNISIG, 2009a) sequence, as if no fault were introduced.

When two faults are applied at the same time, the safe reaction should be recorded. In most cases a fault in odometer is recorded in the JRU and the emergency brake shall be applied through the TIU interface. The safe reaction shall be analyzed prior to the execution of the test, and the appropriate outputs of the system are defined to validate the test. Thus, a sequence with the correct format (UNISIG, 2009a) is executed and analyzed, making the fault injection compatible with the (UNISIG, 2009c) test system.

Fault Injection in BTM

The BTM safety figures imply the use of fault tolerance of one so at least a two-out-of-three voter is used. Thus the implementation of the BTM can be done with TMR architecture, independent of the ERTMS/ETCS Kernel, or as part of the Kernel voter. Manufacturers can choose the architecture, but the information related to balises shall be interchanged.

The saboteurs are placed in the communications that constitute the voter, in the BTM part or in the Kernel part. The saboteur is controlled such that it corrupts the information related to the balise.

Thus, depending on the implementation, the test is done at the BTM level or at the on-board ERTMS/ETCS level. In this later case, any sequence that contains a balise reading can be employed. Again, it is executed three times, once without any fault, a second time with a fault in one branch of the voter, and the third time with faults in two branches but with no relation between them, as they cannot cause the same error. If the voter receives the same erroneous information it will process the wrong information as a majority, bypassing the voter safety, even though the inconsistency of different information may be detected in upper levels.

The procedure is similar to the odometer case. Prior to the test, the safe reaction shall be analyzed and obtained from the requirements. With this information the original sequence of (UNISIG, 2009a) is modified, and the outputs are also modified. In the validation of safety and the availability functionality, in the sequence there are inputs, outputs, and faults. The faults have to be introduced into the documentation, and the outputs shall be modified.

CONCLUSION

The conclusions extracted in this chapter are summarized in the following points:

- The architecture of the on-board ERTMS/ETCS is mainly determined by safety standards, interoperability standards and manufacturing requirements. For example, according to safety standards EN50128, EN50129 and IEC61508, systems performing SIL4 safety-related functions shall have fault tolerant architectures. This requirement introduces new functionality (e.g. majority voting among redundant processing lines) that shall be tested during the validation stage.
- On-board ERTMS/ETCS tests are defined by employing the black box testing technique, which enables the validation of the system performance as a whole.
- Fault injection techniques look for introducing faults in the system with the objective of exciting the internal functionality of fault tolerant systems. There are several

methods spanning from the design stage to the validation stage, that vary in the injection procedure and consequently present different characteristics (e.g. controllability). The technique of choice for facilitating the safety assessment is one that is based on communication saboteurs as the effect at the component level of any internal fault can be emulated by a communication error (corruption, deletion, delay, etc.).

- Currently, interoperability standards define the test setup and test procedures for the on-board ERTMS/ETCS as a whole. In order to test the internal functionality (e.g. communication checking, voting results or fail-safe state) of the SIL4 odometer and the BTM subsystems and hence facilitate the safety assessment, faults can be injected in the communications by saboteurs.

REFERENCES

Arlat, J., Crouzet, Y., Karlsson, J., Folkesson, P., Fuchs, E., & Leber, G. H. (2003, September). Comparison of Physical and Software Implemented Fault Injection Techniques. *IEEE Transactions on Computers*, *52*(9). doi:10.1109/TC.2003.1228509

Arlat, J., & Moraes, R. (2011). *Collecting, Analyzing and Archiving Results from Fault Injection Experiments.* Paper presented at the 5th Latin-American Symposium on Dependable Computing.

Baraza, J. C., Gracia, J., Gil, D., & Gil, P. (2005, December). *Improvement of Fault Injection Techniques Based on VHDL Code Modification.* Paper presented at the Tenth IEEE International High-Level Design Validation and Test Workshop.

Benso, A., Bosio, A., di Carlo, S., & Mariani, R. (2007, September). *A Functional Verification based Fault Injection Environment*. Paper presented at the 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.

Blanc, S., Bonastre, A., & Gil, P. G. (2009). Dependability assessment of by-wire control systems using fault injection. *Journal of Systems Architecture*, *n.d.*, 55.

Carreira, J., Madeira, H., & Silva, J. G. (1995, September). *Xception: Software Fault Injection and Monitoring in Processor Functional Units.* Paper presented at the 5th IFIP Working Conference on Dependable Computing for Critical Applications.

CENELEC EN 50128:2002. (2002) *Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems*.

CENELEC EN 50129:2003. (2003) *Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling*.

CENELEC EN 50159-1:2001. (2001) *Railway applications. Communication, signalling and processing systems. Safety related communication in closed transmission systems". EN 50159-2:2001 "Railway applications. Communication, signalling and processing systems. Safety related communication in open transmission system"*.

Dhahbi, S., Abbas-Turki, A., Hayat, S., & El Moudni, A. (2011). *Study of the high-speed trains positioning system: European signaling system ERTMS / ETCS*. Paper presented at the 4th International Conference on Logistics.

Dutertre, J., Fourniery, J., Mirbaha, A., Naccachez, D., Rigaud, J., Robissony, B., & Triay, A. (2011). *Review of Fault Injection Mechanisms and Consequences on Countermeasures Design*. Paper presented at the 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era.

Fidalgo, A. V., Alves, G. R., & Ferreira, J. M. (2006). *Real Time Fault Injection Using Enhanced OCD – A Performance Analysis*. Paper presented at the 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.

Idirin, M., Aizpurua, X., Villaro, A., Legarda, J., & Melendez, J. (2011, March). Implementation Details and Safety Analysis of a Microcontroller-based SIL-4 Software Voter. *IEEE Transactions on Industrial Electronics*, *58*(3). doi:10.1109/TIE.2010.2062471

IEC EN 61508 parts 1-7: 2003. (2003) *Functional safety of electrical/ electronic/ programmable electronic safety-related systems*.

Jinfu, C., Yansheng, L., & Xiaodong, X. (2007) *Testing Approach of Component Security Based on Fault Injection*. Paper presented at the 2007 International Conference on Computational Intelligence and Security.

Martins, M., & Rosa, A. C. A. (2000). *A Fault Injection Approach Based on Reflective Programming*. Paper presented at the 2000 IEEE/IFIP International Conference on Dependable Systems and Networks.

RIAC. (2010) *RIAC's Relaiability Prediction Methodology 217plus™*. Retrieved from: http://www.theriac.org/riacapps/search/?mode=displayresult&id=351

Skarin, D., Barbosa, R., & Karlsson, J. (2010). *GOOFI-2: A Tool for Experimental Dependability Assessment.* Paper presented at the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks.

Stamenkovic, B. B., & Dersin, P. (2009). Availability Assessment of ALSTOM's safety-relevant trainborne odometry sub-system. In Martorell, S. (Ed.), *Safety, Reliability and Risk Analysis: Theory, Methods and Applications 2009*. London, UK: Taylor & Francis Group.

UNISIG, (2007a) Subset036, *FFFIS for Eurobalise*, Issue 2.4.1.

UNISIG (2007b), Subset 085, *Test Specification for Eurobalise FFFIS*, Issue: 2.2.2.

UNISIG, (2009a) Subset076, *ERTMS/ETCS Class 1, test plan*, Issue: 2.3.1.

UNISIG (2009b), Subset 091, *Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2*, Issue 2.5.0.

UNISIG (2009c), Subset 094, *Functional Requirements for an on board Reference Test Facility, Issue 2.0.2*.

UNISIG, (2010) Subset026, *System requirement specification*, Issue 3.0.0.

Vinter, J., Bromander, L., Raistrick, P., & Edler, H. (2007). *FISCADE - A Fault Injection Tool for SCADE Models*. Paper presented at the 3rd Institution of Engineering and Technology Conference on Automotive Electronics.

Chapter 7
Impact of Electromagnetic Environment on Reliability Assessment for Railway Signalling Systems

Iñigo Adin
CEIT and Tecnun (University of Navarra), Spain

Jaizki Mendizabal
CEIT and Tecnun (University of Navarra), Spain

Jon del Portillo
CEIT and Tecnun (University of Navarra), Spain

ABSTRACT

The electromagnetic interferences (EMI) are threats that affect the reliability of the railway signalling systems. Consequently, the identification of the reliability requirements dependent on environment conditions is a major issue for signalling systems designers, and therefore for evaluators, and testing and certification bodies. Signalling systems work in the complex and heterogeneous railway environment, where low power electronics have to work together with high voltages and currents from trains and railway infrastructure. This chapter presents the relationship between the railway electromagnetic interoperability and the reliability assessment by analyzing the signalling systems and the associated inter-dependencies with other components of the rolling stock. It is composed of two main sections; the first gathers an exhaustive state of the art approach to the issue of electromagnetic interoperability and railway industry. This subsection steers towards the combination of electromagnetic interferences and the signalling systems present in the rolling stock noise environment. That is the basis of the second section that finally sets how to establish the reliability requirement for a communication path in this environment. This requirement is established because of the electromagnetic noise environment, as well as the radiated and conducted fields, which are a combination of all the surrounding threats a focused railway system has to face. It also depends on the modulation of the communication signal under study.

DOI: 10.4018/978-1-4666-1643-1.ch007

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

A wide range of definitions can be found for the term "reliability", but the one proposed by the IEEE in 1990 is (IEEE Std 610.12, 1990):

"Reliability is the ability of a system or component to perform its required functions under stated conditions for a specific period of time"

The thesis of Vromans in 2005 (2005), based on that statement, discussed the improvement of railway reliability in the form of the development of tools and guidelines helpful in the construction of reliable timetables for railway traffic organization.

And even if the goal of the study of the reliability of service is not exactly in the line of the scope of this chapter, which is mainly focused on the reliability requirement extracted from the electromagnetic interoperability, it proposed an interesting list of definitions in direct correlation to the term "reliability". The one which would be directly related to interoperability and reliability is "robustness":

"The robustness of a railway system indicates the influenceability of a system by disturbances. A robust railway system can function quite well under difficult circumstances. When a railway system is not robust, small external influences cause large delays which propagate quickly throughout the system in place and time".

In this definition, "disturbance" is the key term. Which are the disturbances of the railway systems in terms of interoperability? The railway electromagnetic noise environment. This is why this chapter collects and defines all these threats for the rolling stock environment. The first section of this chapter starts with the state of the art of the electromagnetic compatibility issue for the railway industry. To ensure compliance and to satisfy safety and reliability requirements, it is necessary that rail vehicles and railway on-board and track equipments are rigorously designed and EMC tested prior to service introduction. Nevertheless, the lack of an ultimate electromagnetic compatibility test setup and procedure with the consideration of the specific worst cases is a setback in terms of cost, time and, of course, reliability. The norms and standards that rule these issues are completed in this chapter with the scientific community contributions for a global understanding of the present situation.

Consequently, the threats for the rolling stock are presented. The railway noise environment is mainly composed by the following noise sources/ EMIs:

- Thermal noise
- Spot signalling systems
- On-board communication systems
- Track communication systems
- Electromagnetic fields generated by traction systems
- Overhead lines noise and discontinuities in the contact between the catenaries and the pantograph
- Electrical substation noise

The combination of these threats establishes the magnetic field spectrum with the steady state and the transient conditions for the rolling stock. Obviously, depending on the threatened system studied in each case, a specific frequency band has to be considered and the signal and noise spectrum around that frequency delimits its reliability requirements. The spectrum characterization is also schematically presented in this chapter, in the second section, through an example of spot signalling system. A similar analysis would be applicable for any other signalling system for railway applications.

The last section of this chapter explains the relationship between the accomplishment of the reliability requirement of a signalling system and the noise received by this system. The procedure presented in that section leads to the quantification

of the Bit Error Rate (*BER*), parameter related to the reliability characteristic of the system, through the Signal to Noise Ratio (*SNR)*, obtained from the limiting spectrum and the modulation used for the communication. This procedure is presented for a specific signalling system, in a real railway working environment.

As a result for this chapter, the reliability requirement is then here linked to the electromagnetic interoperability and the system design. Therefore, testing and certification bodies as well as designers and evaluators will have a better control of the influence of the electromagnetic interferences, and consequently of the interoperability, from the early steps of the system definition.

RELATIONSHIP AMONG ELECTROMAGNETIC ENVIRONMENT, DESIGN CONSTRAINTS, COSTS AND RELIABILITY

Currently rolling stock electromagnetic emissions are a major concern for train manufacturers, signaling systems developers and railway infrastructure operators (ERA EMC Report 2010). Available harmonized EMC standards (EN50121-2, EN50121-3-1 and EN50121-3-2 (EN50121, 2006)) do not completely address interoperability issues caused by rolling stock interferences with signaling systems (GSM-R, track circuits, spot signaling systems such as BTM or LTM). Moreover, these standards do not cover representative worst-case conditions derived by transients in the rolling stock behaviour typically generated by feeding and track circuits' discontinuities.

On one hand, this situation causes an important waste of time and resources for train manufacturers when integrating rolling stocks and signaling systems, and also on already tested trains, occasionally problems may still arise. Then, not only the responsibilities but also the technical solutions are not straightforward. The duration of the field testing employed to solve this kind of problems and to go through the certification process may vary between 3 months and 12 months. And the cost of the complete process may vary from 25k€ to 1,5M€ (ERA EMC Report 2010).

On the other hand, railway infrastructure operators suffer the railway infrastructure availability reduction caused by the rolling stock electromagnetic incompatibility with the safety critical signaling systems. The previously commented problems might cause an estimated reduction of 10% of the availability in the most crowded lines.

This is the reason why an extended study of the state of the art of electromagnetic compatibility for railway environment is here below presented, to set the starting point for this chapter. That is the base for further analysis of interoperability issued and the related assessment of reliability requirements.

State of the Art

The state of the art of this topic is composed of four subsections which are complementary sources. The first one shows the official norms and standards applied for electromagnetic compatibility in railway applications. The second one describes the contributions made by the scientific community on the specific topics here interrelated. The third reference is dedicated to the EMC and interoperability studies and reports, whereas the fourth is focused on the European funded projects which studied the "Electromagnetic compatibility between rolling stock and rail-infrastructure encouraging European interoperability". The following list presents the overview of this issue:

1. Norms and Standards (EN50121, 2006; EN50238, 2003; EN50338, 2001; EN50215, 1999)
2. Scientific Community contributions:
   a. EMI and telecommunications services (York 2002) and (York 2004)

   b. EMI and GSM-R (Slimen and Deniau 2008)
   c. EMI and spot signaling systems (J. Del Portillo 2008)
   d. EMI and track circuit (Niska 2008)
   e. EMI and power supply system (R. Dolecek 2007)
   f. EMI and contact discontinuities (Tellini 2008)
3. EMC and Interoperability studies and reports (ERA EMC Report 2010)
4. European projects (Railcom, 2008) (TREND, 2007)

Norms and Standards

In order to ensure that any type of equipment is designed to perform correctly as close as possible to its environment, the European standards making body CENELEC (European Committee for Electrotechnical Standardization) has been mandated to produce standards for use with the European EMC Directive. ETSI (European Telecommunications Standards Institute) is the mandated standards body for telecommunications equipment.

For railway systems, the only approach standardised to control the emission levels of the railway vehicles is defined by the standards EN50121 (2006). However, these standards aim to control the EM pollution generated by the railway infrastructures and vehicles to the outside world.

Consequently, no methodology is proposed to characterize the EM environment present on board the trains. In the EN50121 (2006), the EM emissions of the whole railway system are measured at 10 m from the middle of the tracks. The measurement protocol is specified for four frequency bands

- 9 kHz-150 kHz
- 150 kHz - 30 MHz
- 30 MHz - 300 MHz
- 300 MHz-1 GHz

The purpose of these standards is to protect the neighbourhood of railway infrastructures, and as mentioned before, no recommendations are specified to characterise the EM disturbances on board (above, inside and under the train). This can be an issue especially where new and future sensitive systems can be placed or designed for.

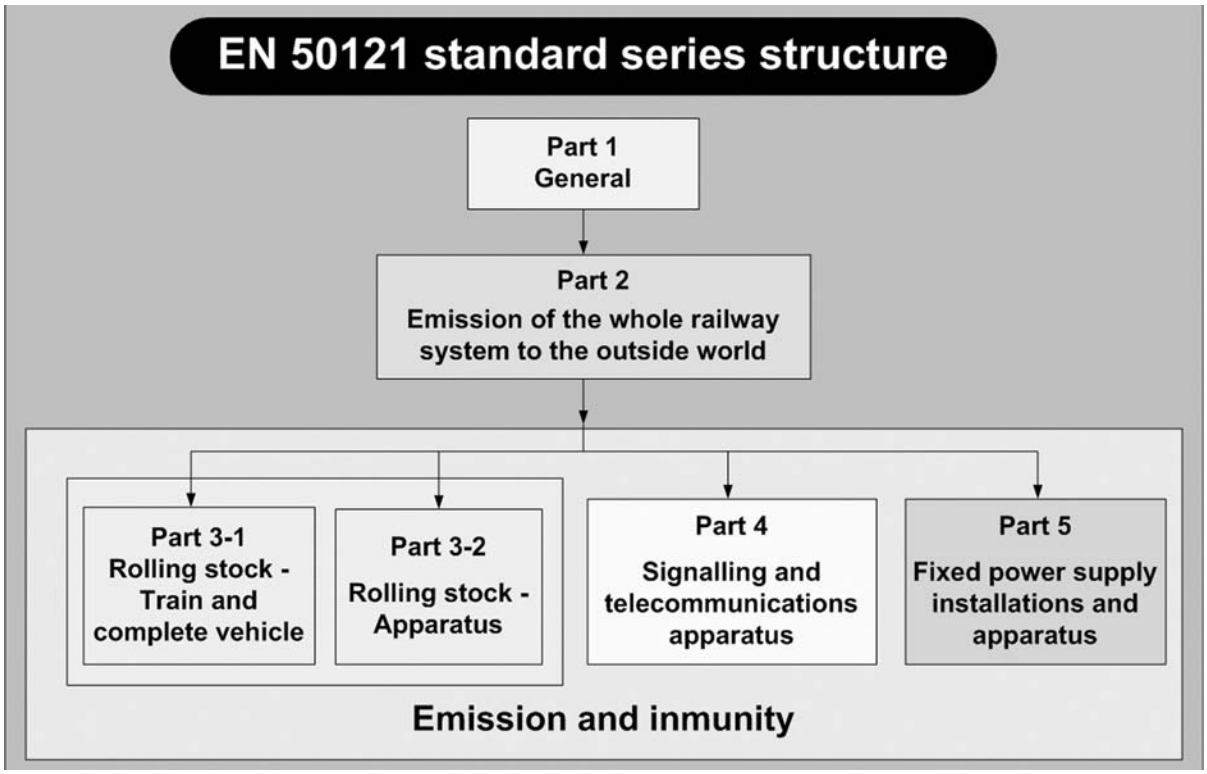
The Interoperability Directives require EMC issues to be addressed, both internally (between trains and infrastructure) and externally (between the railway as a system and the outside world). However, it is recognized that neither the Technical Specifications for Interoperability (TSIs) already established, nor the High-Speed Rail Interoperability Directive, nor the forthcoming TSIs for Conventional Rail Interoperability address completely the complex railway EMC issues.

Currently, only the EN50121 (2006) standard is indicated in the EU directives in relation to the demonstration of Electromagnetic Compatibility of rolling stock. Nevertheless, other norms are focused on interactions between the rolling stock and other components of the railway infrastructure, and hence, some information can be extracted from them in relation to the disturbances of the systems integrated in the rolling stock.

The first statement of this state of the art is that the norms and standards approved up to the date of publication of this book (2011) have areas of improvement to cover all the range of disturbances and interoperability issues in the railway environment. The overview of these norms and standards, here below, highlights their weak spots and the areas of improvement.

The EN50121 (2006) series is dedicated to Electromagnetic Compatibility for railway systems. It generally deals with emission and immunity (radiated and conducted) between the rolling stock and itself, and between the rolling stock and other parts of the railway environment. More specifically, the standard EN50121 (2006) gathers the mandatory requirements for new railway applications, which includes traction rolling stock, mainline vehicles and urban

Figure 1. EN50121 standard series structure



transit, all requiring testing for compliance with the European standards. The EN50121 (2006) standard addresses three main electromagnetic interferences:

1. The effect on the environment surrounding the railway.
2. The effect on communications and signaling equipment.
3. The capacity of the installed equipment to remain unaffected by environment (including weather effects) in which it operates.

This series is composed of 5 parts, as shown in Figure 1.

However, there are some gaps that should be covered: Interactions between the rolling stock and several items of safety critical equipment and controls on the railway. These include train detection systems, signaling systems (BTM, ASFA, LTM, LZB or EBICAB), warning and automatic train control systems, telecommunications equipment (both land and air based) or any equipment operated on or near the rails.

Another area to be improved is the limit of the electromagnetic spectrum which sweeps from 9 kHz to 1 GHz. A full coverage is preferable to consider any harmful interference for a global harmonization.

Finally, the limits do not ensure a safe operation. An improvement is needed in the repeatability, measurement reliability and affordability. This improvement, moreover, should be faced from a European Union perspective, since there can be different legal interpretations of the EMC Directive within different member states.

Besides EN50121, there are three other standards, EN50238, EN50388 and EN50215 that indirectly affect rolling stock electromagnetic emissions and so should be considered in this state of the art section.

EN50238 (2003) directly addresses the main issue exposed in its title, but primarily deals with processes for demonstrating compatibility rather than defining system parameters.

EN50238 (2003) standard and national standards partially cover interactions between the rolling stock and the train detection systems. One idea of the utility of the EN50238 (2003) can be obtained by the statement provided in the 67575_ERA_EMC_final_Report (ERA EMC Report 2010). It says "*EN50238 is currently under revision/expansion however, the only part which is officially published in the journal (and hence universally applicable) is Part 1. Part 1 deals with general procedural aspects, responsibilities and techniques but contains no specific engineering data that may be used in a demonstration of compatibility".* Part 2 and Part 3 are related to track circuits and axle counters, respectively.

EN50388 (EN50388) standard mainly deals with power supply and includes aspects of line resonance effects. Whilst not specifically applicable to the demonstration of Electromagnetic Compatibility, the resonances in the system may alter emissions from the rolling stock in a potentially destructive way.

Finally, the EN50215 (1999) standard is intended to be used as technical instructions for the processing of tests which may be needed for demonstration of certain technical requirements where they are relevant. Specifically, section 9.15 deals with electromagnetic compatibility tests, but it directly references EN50121-3-1 (2006), EN50121-3-2 (2006) and EN50238 (2003), and so shares their characteristics, properties, but also their weak spots.

From a more generic point of view, in Europe, there are two general European Directives; the Physical Agents Directive (electromagnetic fields) 2004/40/EC and the Electromagnetic Compatibility Directive 2004/108/EC which applies to all electrical equipment. These two directives include requirements and standards that explore conducted, induced and radiated phenomena. Some synergy between any compatibility demonstration to these directives and technical aspects of demonstrating compliance on the railway would be expected. However, present practices do not show much synergy. Concepts like the compatibility margin or controlled environment, which should be taken into consideration by any designer are not addressed in any part of the EN50121 (2006) standards' series.

Scientific Community Contributions

The scientific community has gone one step forward describing and analyzing concrete situations of electromagnetic interferences (EMI) between the rolling stock and a wide range of systems and applications. These publications have highlighted several problems of interoperability that are not covered by the previously mentioned standards. These are listed and summarized here below and they should also be considered as input data for the building of the noise environment, in the second part of this chapter, where the reliability requirement is being deduced for any design for railway applications.

EMI and Telecommunication Services (York 2002) and (York 2004)

These two studies carried out by York EMC Services Ltd. in 2002 and 2004 present a deep analysis of potential electromagnetic interferences generated from the railway vehicles and infrastructure.

The first one was planned because of the concerns raised in the early stages of the EN50121 (2006) series definition about radio frequency emissions from railways and their potential to interfere with the commercial radio services and other equipment such as information technology (IT) equipment operating adjacent to the railway lines. An extract from the conclusions written in the generic "Executive Summary" and in the conclusion chapter of the document is here reported:

"The findings of this study have implications for planned or existing buildings in which IT equipment will be used, where the buildings are situated very close (i.e. less than 10 m.) to electrified railway lines. There is a significant probability that the passing trains will interfere with PC monitors that are only a few meters away from the lines.

It is proposed that reduction of EM emissions from the railway can only be achieved by a progressive reduction in emission limits defined by the standards; this would be expected to have an effect over a number of decades.

[...] The interference levels specified in EN 50121 do not, in the worst case, provide adequate protection for current users of the AM, FM or for users of the DAB bands."

After this first measurement campaign came the second one, in 2004, addressed to OFCOM (Independent regulator and competition authority for the UK communications industries). It describes an improved method for the measurement of radiofrequency emissions from railways. From the eight recommendations proposed in the conclusions of this study, three of them should be highlighted:

1. The maximum emissions are not necessarily seen when the train is passing the measurement position as in the case of an overhead power supply system.
2. Transient and broadband emissions may be present as a result of pantograph arcing on high speed lines and also as a result of arcing in brushed motors.
3. Emissions in a single frequency range may be both harmonic (narrowband) and broadband in nature from a single vehicle.

With regard to the interoperability and the reliability issue of railway signalling systems dealt in this chapter, a qualitative conclusion may be extracted from these two studies carried out by York EMC Services Ltd: the interference levels measured from the railway vehicles has serious implications in the reliability of the Communication and IT systems placed less than 10 meters away from the track. That implies, undoubtedly, that this has at least the same serious implications for the Communication and IT systems placed inside and below the train. Even if the worst case is not necessarily seen by the characterization equipments when the train is passing, the levels are still above the limits that assure the complete fulfillment of the reliability value.

EMI and GSM-R (Slimen & Deniau 2008)

This paper deals with the electromagnetic characterization of the railway's environment on a moving train, in the frequency band extended from 300 MHz to 1 GHz. This specific frequency band is the upper band determined by the EN50121 (2006) standard for the measurements protocol, and this is the key for the European Railway Traffic Management System (ERTMS). The traffic controlled by this system has to be ensured by means of the Eurobalise systems and the GSM-R. The latter is a continuous communication system (voice and data) between the train and the control centre, whereas the former should allow the trains to be located and punctual data to be transmitted from the control centre to those trains. These two systems have to be sufficiently robust for the EM disturbances produced in the railway environment to guarantee the safety of the railway transportation.

The authors of this publication were worried by the limitations of the purpose of the standards: control the emission levels of the railway vehicles and infrastructures. Indeed, no recommendations are specified to characterize the electromagnetic disturbances on board (above, inside and under the train). In order to determine the immunity levels that the on-board systems have to verify

to correctly operate in the railway environment, Slimen et. al. proposed to perform on-board EM measurements to characterize the interferences which can affect the on-board telecommunication systems (specifically above the train to focus on the interoperability of the GSM-R communication system). Comparisons between measurements in time domain and frequency domain were performed under different operating conditions. The conclusion from this paper is that the time domain measurements method requires a huge post processing capability, but is well appropriate to collect the whole spectrum of emissions with their transient episodes (which can be crucial for punctual reliability failures in trains). The frequency domain approach was also appropriate but could "slightly underestimate the EM noise levels". The election of the time domain approach or the frequency domain approach should be analyzed in each case.

EMI and Spot Signaling Systems (J. del Portillo 2008)

This publication captured the state of the art (up to 2008) concerning the electromagnetic compatibility for the on-board spot signalling. As seen for the ERTMS systems, trains require these systems to be supervised and controlled. Prior to the establishment of this European unified system, each member state, or group of states could have their own signalling system (i.e. ASFA for Spain). The conclusions of this paper set that although major research have been carried out, currently neither the standards nor the state of the art detail the characteristics of the electromagnetic environment for the on-board railway spot communication systems. For the ASFA and the BTM systems the main possible origin of problems is the electromagnetic fields generated by the train itself. The common mode currents generated in the traction system are mainly located in the kHz range and could affect the operation of the ASFA system. The train, for the common mode currents, behaves electrically like a resonant tank and the over-damping response has a resonant frequency that interferes with the ASFA signal. Furthermore, the contact discontinuities between the pantograph and the catenaries also introduce an over-damped response, of which its tones may affect the performance of both spot communication systems.

This is an evident example of electromagnetic interference issue between two systems installed in the same train. Moreover, in that case, the train itself has worsened the electromagnetic fields generated by the traction system and the contact discontinuities between the pantograph and the catenaries.

Consequently, besides the collection of all the signals present in the environment of the train and which form the electromagnetic noise environment, these effects also have to be considered for the input data needed to deduce the reliability requirement, in the second part of this chapter.

EMI and Track Circuit (Niska, 2008)

This study made a contribution towards an understanding of the EMC and EMI characteristics of the signalling and detection infrastructure of the railway system. These characteristics often lead to failures resulting in train delays. The study analyzes, in detail, the failures in the railway environment and reporting systems. Data collected in this study showed that most of the failure causes are related to EMC problems. The fault reporting system is not configured to identify the failure causes as EMC or EMI problems. Therefore, this has been investigated to identify the cause so that corrective actions can be initiated to restore the system to an operating condition.

The research study has helped in understanding the function of railway signalling and detector installations from an EMC and EMI point of view. The knowledge generated is of assistance in designing new signalling and detector equipment which have a higher level of reliability, leading to a smaller number of failures and EMC problems.

This master thesis employs a complex model of the railway system and measurements for the characterization of EM interferences in detector infrastructure of railway systems. At last, two conclusions are taken from this study:

- Demonstrably high transients occur in the system and can provide enough energy to interfere with the functionality of subsystems and hence the reliability of the system.
- The ground is vital in the measurement of any EMI and EMC episode: The recorded ground frequency in the measurements is 8 1/3 Hz, which is half the frequency that is used in the railway, 16 2/3 Hz and has no relationship to any other systems in the railway

EMI and Power Supply System (Dolecek, 2007)

This paper analyzes, through a modelization, the transient effects for the power supply systems from electromagnetic compatibility viewpoint. The main problem of the galvanic coupling of the contact line in a feeding station is the transient effect during short-circuits. As explained in this publication, this effect can arise during failure in a traction circuit. Consequently, the protection of these circuits is crucial to avoid any interference not only in the feeding circuit of the train, but also interferences in the on- board equipment due to the fields originated by this coupling (The signalling systems are especially susceptible to damage but the traction circuits can also suffer due to these current peaks). The solution proposed by the authors of this publication is to reduce harmonics of the supply signal and the design of protection from EMC point of view.

EMI and Contact Discontinuities (Tellini, 2008)

As presented in the abstract, this paper focuses on the measurement of electromagnetic interferences from high power systems characterized by fast and intense electromagnetic transients. The contact discontinuities presented here are those encountered in the operation of rail electromagnetic launchers, but the conclusions can be extrapolated to railway system. The characterization has been done during the emissions produced during the plasma formation at the contacts between the brushes and the rails. The experimental results of Tellini et al lead to the conclusions that the most significant emission phenomena occur in the frequency range of the tens of MHz. Unfortunately, those emissions could affect the on-board communication systems in trains.

EMC and Interoperability Studies and Reports

The last source of information to study the issue addressed in this chapter are the specific studies and reports published.

The main one is the document "67575_ERA_EMC_Final_Report", where key information to understand the state of the art in EMC and interoperability is found. The European Railway Agency approved this study entitled "EMC for European Railways" developed by Lloyds Register Group, in June 2010.

It is focused on the examination of the processes, procedures and methods ensuring electromagnetic compatibility between rolling stock and infrastructure in the 27 members of the European Railway Area. In particular, it analyzes the demonstration of EMC of the rolling stock with the requirements of operating infrastructure.

First, the ideas were developed from reviews of relevant standards and prior work in this area e.g. "Railway applications – Interference limits of existing track circuits used on European Railways (PD CLC/TR 50507] and Safety Regulations and Standards for European Railways (NERA, 2000). And later, information was sought among the 27 members of the ERA on specific interactions and demonstration of compatibility processes, mainly

for the train detection systems, line side equipments, energy supply and neighbouring radio frequency systems.

The document revealed the following issues:

- The two technical standards that should demonstrate electromagnetic compliance of rolling stocks do not ensure compatibility between the rolling stock and the signalling systems, nor with telecommunication services. Nor can they ensure the safety mode operation.
- Telematics predominantly concerns the tracking and logistics associated with freight, particularly containerized freight. Cargo is identified by passive or active radio-frequency tags. These are considered to be a separate component of the railway and are assessed by a separate TSI (commission Regulation EC No 62/2006 December 2005).
- In some countries (e.g. Austria, Belgium, etc.) additional requirements are set to limit radiated emissions that could affect telecommunication services (e.g. 80 MHz, 160 MHz, 450 MHz, 900 MHz, 2,4 GHz, etc.)
- In some countries (e.g. France, Greece, etc.) human exposure to electromagnetic fields (EMF) is considered part of the Electromagnetic Compatibility considerations for train acceptance.
- Actual timescale of the testing and operational trials is between 3-4 months in some countries (e.g. Ireland, Germany) and 12 months for others (e.g. Czech Republic). Cost of the certification process varies between 25 k€ (Poland) and 1.5 M€ (Czech Republic).

European FP7 Projects

The RAILCOM project was a partly EU funded Specific Targeted Research Project (STREP) with 17 consortium partners. The FP7 project started in December 2005, with a funding of approximately 2.45 M€, and finished in 2009. The project consortium included rolling stock and system manufacturers, infrastructure owners, train operators and research institutes: Alstom, AnsaldoBreda, Bombardier, CˇD, CNTK, DB AG, INRETS, Movares, Nitel, RFF, SBB, Siemens, SNCF, TU Kaisers lautern, UIC, UNIFE and VUZ.

The objectives of the project were:

- Development of harmonised methods for the definition of interference limits for train detection systems on the TEN-T railway network (Trans-European Network for Transport), and corresponding methods for testing of rolling stock.
- Characterisation of the electromagnetic environment of railways for communication systems, with correlation between emission and operating conditions of the system.

As a result, the project addresses improvements on two specific railway interfaces between rolling stock and infrastructure on the TEN-T railway network. These results were provided to CENELEC for the ongoing standardization process in these two fields:

- Compatibility between conducted interference from rolling stock and track circuits for train detection.
- High frequency interference in communication systems between rolling stock, infrastructure, employees and passengers.

RAILCOM project incorporated some very important advances in the EMC and interoperability. However, there are some key points not addressed, which have been proposed by the project TREND. It is the research project of a subsequent consortium in the next interoperability topic oriented to railway sector European projects call (FP7-SUSTAINABLE SURFACE TRANSPORT

(SST)-2011-RTD-1). The main points proposed by the TREND project, are explained as follows:

Regarding the current harmonics generated by rolling stocks influencing track circuits, RAILCOM progressed beyond the available standards proposing a new methodology. This methodology was passed to the pr15360 / EN50238-2 (2003). Nevertheless some points should be further improved:

For example, the harmonics generated by the rolling stocks affecting the track circuits are multiplied by a multiplication factor which can be applied for non-synchronised harmonics. Whether harmonics are synchronised or not has to be judged for each individual frequency band. RAILCOM approach in this case may lead to a situation of overestimating the value of harmonics, which would increase the cost of countermeasures, or underestimating which may led to potentially unsafe situations. An alternative approach would be the identification of the representative worst case situation and the definition of a test setup for making these situations repeatable.

Regarding the interactions between the rolling stock emissions and the spot signalling systems like BTM and LTM, only BTM was covered (LTM are neither analysed nor mentioned). Moreover, the test setup proposed to analyse the immunity of the BTM was only based on the vertical magnetic field and a relatively narrow band antenna. Unfortunately, the reality is quite more complex. The actual most probable EM environment will contain transversal, longitudinal magnetic fields and huge electric fields located from the KHz to the tens of MHz frequency range. Manufacturers experience reveals a 10 dB noise reduction by providing electric shielding. Therefore a more complex test setup is required. Besides, the handling of EM transients, no representative worst case identification for BTM nor LTM was made.

Regarding GSM-R, the RAILCOM project developed an excellent work. However, this work was not concluded and in the final RAILCOM meeting future work was outlined. The points to accomplish were:

- Complete characterisation of the time characteristics of the transient disturbances for GSM-R produced by the catenaries-pantograph sliding contact
- Proposition of a prediction method of the *BER* induced by the transients observed on board as a function of the repetition rate of the transients
- Proposition of a laboratory testing method to control the immunity of the GSM-R communications against EM disturbances representative of the in situ conditions (which should cover permanent and transient disturbances, simultaneously)
- Proposition of a methodology to preliminary verify the conditions required to guarantee the quality of the communications

Regarding the influence of rolling stock emissions with broadcasting or telecommunication services no improvement at all was done. No assessment of the measurement test setup, methods or conditions was made.

In the conclusion of this project, the SST.2011.2.5-1 call in the transport topic, enclosed in the area of Interoperability and Safety proposes the "Harmonization of freight and passenger Rolling Stock approval tests for electromagnetic compatibility (EMC)". In that frame, several proposals were launched and the TREND project was accepted for funding. This project's title is "Test of Rolling stock ElectromagNetic Compatibility for cross-Domain interoperability" The starting date of this project is November the 1st 2011, and the expected duration is 30months (TREND, 2007).

Conclusions of the State of the Art

The conclusions derived from the exhaustive state of the art analysis here presented are summarized in the following four points:

1. No specific recommendations are specified in the standards to characterise the EM dis-

turbances on-board the trains (above, inside and under the train).
2. The certification process varies enormously in time and cost across Europe and, besides, extra requirements are introduced in some countries.
3. Demonstration of permanent electromagnetic compatibility of the rolling stock in all the possible conditions is not feasible with the current standards available.
4. The research community has also made significant improvements in understanding the physical phenomena that are behind the EM incompatibilities between the rolling stock and other systems (track circuit, signalling and telecommunications services). Unfortunately, the research areas have been isolated and a cross-domain approach has not yet been employed.
5. RAILCOM FP7 project has made important research in the fields of EM emissions of the rolling stock that affect GSM-R, track circuits and BTMs. However this is only a starting point because only some issues of the overall problem have been partially addressed.

This is the starting point of the studies for the future years. TREND and other FP7 funded projects will go some step forward the harmonization of the electromagnetic interoperability for railway systems.

The second section of this chapter summarizes all the signals present in the railway noise environment, which leads to the characterization of the electromagnetic environment.

ELECTROMAGNETIC NOISE ENVIRONMENT CHARACTERIZATION

The reliability of the communication and signalling systems present in the rolling stock depends highly on how noisy the received signal and the communication path are. Each system has its own allowed maximum noise level and so these systems are designed depending on the received signal level and the noise ratio. This is why the definition of the magnetic field spectrum is so crucial: it allows the designer to define the system characteristics.

As shown in the first section of this chapter, EMC standards do not cover completely the interoperability issues analyzed in some research works. These works have defined several noise processes that can be generated by different sources in the train or by systems in its environment. The most common ones are listed below.

External sources of interferences:

- Neighbouring railway systems
- Industrial plants which disturb the electricity supply network
- Trackside radio stations
- Radar sets at airports or on aircrafts

The interferences caused by the train itself are due to:

- Trackside equipment
- Auxiliary power supply converters
- Traction return current

Independently of the system studied from the EMC perspective, most practical EMC problems have a common background which is usually irrespective of the application sector. However, the interference from currents circulating under the train is the most peculiar EMC hazard to railways. This hazard is manifested in two forms (Cecube):

1. A longitudinal voltage developed along a train that can affect an adjacent track and its track circuits, either with or without the presence of infrastructure track faults.

2. A longitudinal current flowing under the train, when the train interconnects an insulated rail joint that has a broken side-bond.

Obviously, jointless track circuits avoid the problem created by longitudinal current, but for many legacy signalling systems it remains an unrecognized or under-estimated hazard. The hazard analysis is concerned with the risk that the train propulsion control or auxiliary system fails in such a manner that excessive levels of leakage current can enter the running rails. This would be identified by the false activation of track circuit relays from excessive longitudinal voltage or current at the signalling frequencies. If this hazard occurred it could lead to a false signal indication and potentially result in collision.

Proving these risks are acceptably small requires a careful analysis of the return current and earthing strategies employed by the rolling stock. Current harmonics flow the length of the train or auxiliary cabling permit parallel current paths in the vehicle body or rails. Those conducted interferences can even create failure events on a neighbouring track.

Similar reasoning on the effects on the rolling stock would be applied for the radiated interferences affecting the signalling systems, as seen just below. Our belief is that a predictive perspective of the problem can be adopted from the design stage. The next subsection presents the study done for the signalling systems of the rolling stock.

Interferences Definition for a Given Signalling System

The main objective of this subsection is the definition of the electromagnetic spectrum received by a signalling or communication system installed in a train. This is the starting point for the reliability requirement establishment of this work.

A number of different signalling systems exist nowadays. The European ERTMS tends to harmonize the signalling systems, trying to minimize the employ of National railway signalling systems. Among all these systems the following ones can be found: BTM, ASFA, LTM, LZB, EVM, EBICAB, etc. A long list of signalling systems can be found in the bibliography, which proves that this sector needed an urgent homogenization, as the nowadays ERTMS pretends.

In order to characterize the receiver electromagnetic spectrum the main characteristics of each system, such as frequency of operation, bandwidth, modulation and bit rate, has to be determined. Some of these system characteristics that work in the spectrum from tenths of kHz to tenths of MHz. are described as follows:

- BTM receives the information from the balise by means of a FSK signal whose center frequency is 4.234 MHz ±175 kHz and frequency deviation is 282.24 kHz ±7%. Then, in the shift between the two frequencies, the carrier has a continuous phase. The transmitted information mean data rate is 564.48kbit/s. (SUBSET036, 2005)
- The trackside ASFA balises transmits the information to the on-board system via several frequencies between 55 kHz and 115 kHz. Each of these tones is related to a specific command. Therefore, there is not any modulation used in this communication.
- LTM's centre frequency is 13.54MHz ± 30ppm and the data uplink has a pulse-code modulation to provide a data rate of 9.6 kbits/s. The telegrams are transmitted from the trackside to the on-board equipment by means of a Direct Sequence Spread Spectrum (DSSS) and Binary Phase Shift Keying (BPSK) modulation scheme. (SUBSET044, 2004,)
- LZB operates by exchanging telegrams between the central controller and the trains. The central controller`s telegram uses Frequency-shift keying (FSK) signalling at 1200 bits/s on a 36 kHz ± 0.4 kHz and

the train's response telegram performs 600 bits/s at 56 kHz ± 0.2 kHz.

- EBICAB uses also a track-to-train communication and the balises transmits the information to on-board system by means of a 255bit telegrams in a 4.5MHz centre frequency with a 50kbps mean data rate.

Then, the noise in the specific bandwidth of the signalling system must be characterized in order to establish the system requirements. Nevertheless, as mentioned in the introduction of this second section, the noises interfering with a signalling system strongly depend on their location on the on-board traction systems of that train, on the auxiliary power converters, or even on other interfering signalling systems overlapping the threaten frequency band.

The studies and publications referenced in the first section of the chapter showed some interesting results. In this section the frequency of these interferences are threshed and only the parts affecting the frequency of the victim signalling system are maintained.

- (Niska, 2008) Shows the effect of ground frequency in track circuits. This frequency and its harmonics compose the spectrum of the noise generated in this case. All the frequencies are below 150Hz and so have no effect on the victim spectrum.
- (Slimen & Deniau, 2008) Analyzes the bandwidth from 300MHz and 1GHz where the disturbances for GSM-R can be found. These frequencies are much higher than the maximum frequency defined for the signalling systems.
- (del Portillo, 2008) Analyzes the disturbances generated by traction systems and the discontinuities between the catenaries and the pantograph. These two detected noises affect the signalling and communication systems and are generated by the train itself. The former is due to the generated common mode currents. These interferences appear due to the chopping of the input voltage received from the catenaries by using IGBT's at high frequency. The current flowing to the circuit in the three phases is not perfectly balanced and the common mode currents appear. This current, which flows every time through the IGBT switches, follows an over-damping pattern whose main frequency is 60 kHz. The latter usually corresponds to the union of two sections connected to different substations. This absence of contact results in a transient current generated in the power supply which is the origin of interference. A variable magnetic field generated by the return current through the train is received by the signalling system resulting in its failure. That return current has harmonics in the victim spectrum environment.

These systems failures result from the effect of the interferences here highlighted are false track system detections or false alarms. Both noises were characterized, as defined just below, by means of the time domain measurement method.

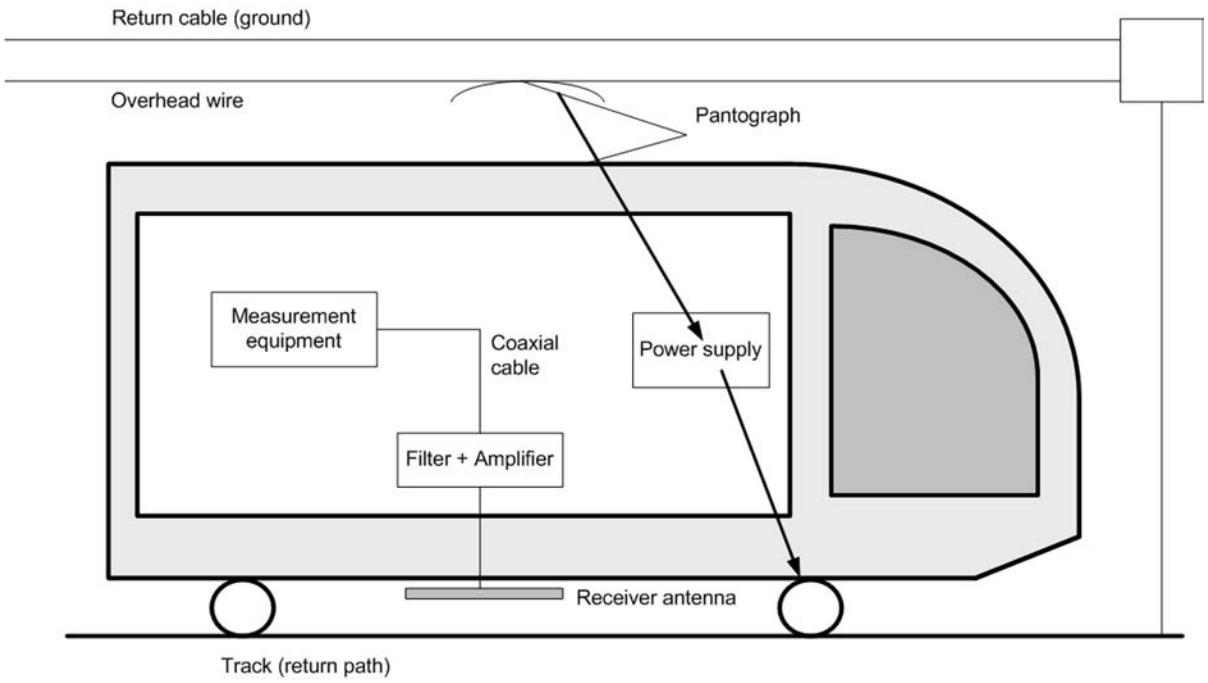
As a summary, for the example of the signalling systems contained in the spectrum from tenths of KHz to tenths of MHz, the sources of interference are the two explained just above. For the determination of the real amplitudes affecting the analyzed signalling system, a physical characterization was performed. That sets the noise and its spectrum influence.

Test Set-Up for Noise Characterization

The following test set-up is established in order to characterize and define all the noises.

First, the determination of the capturing antenna bandwidth and the antenna location are two key points. The antennas have to be designed and need to be located below or above the train

Figure 2. Test set-up for emission measurements



depending on the signals that are expected in the frequency environment.

Then two methods can be used to record the received signals: frequency domain or time domain. The differences between both methods are simplicity against accuracy.

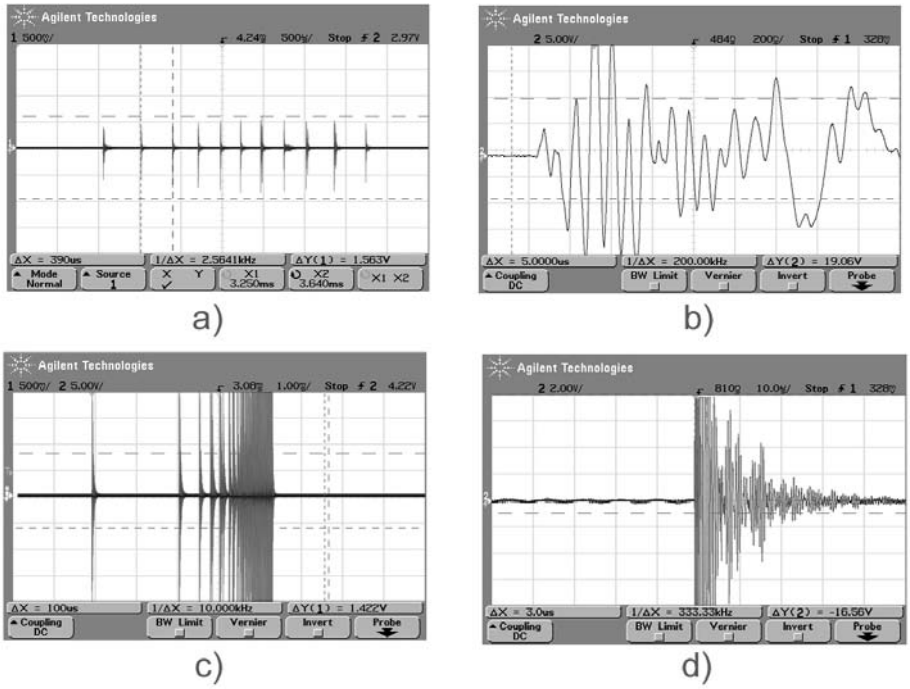
On the one hand, the time domain measurements method is appropriate to collect all the spectrum of transient emissions, but the post processing needed in this case complicates the analysis. On the other hand, by means of the frequency domain measurements the spectrum is acquired without any post processing, but due to the sweep of the measurement equipment, transient events could be lost, making the measurement less precise.

Therefore, if the measurement equipment performance allows storing all the information measured in a given measurement and the post processing program is developed in order to calculate reliably the signal spectrum, this method will be chosen. Nevertheless, if those conditions cannot be obtained, the frequency domain measurement method would be the alternative. One example of a test set-up is the one shown in Figure 2.

By means of the signal captured employing such a test setup, (J. del Portillo 2008) studies the effect of the pantograph operation in the signalling systems. The pantograph was raised and lowered several times and the signals captured by the antenna located under the train allowed to extract the following information:

- The transient events presented a similar pattern in all the measurements carried out in these tests.
- The measured current and the received H-field by the on-board antenna had exactly the same shape. Therefore, it can be confirmed that the current was responsible for the transient H-field.
- Figure 3c shows one transient event duration, which varied between 1 to 10 ms.
- Several peaks composed each transient and their over-damping response or ringing can

Figure 3. Transient voltages measured in the antenna with the oscilloscope



be identified. Figure 3a shows some of the peaks of Figure 3c and clearly presents the shape of the peak and the over-damping response. The separation between peaks is around 100 and 200us.

- The over-damping response of all the peaks is shown in Figure 3b. A high frequency oscillation at the beginning (during 2us) and a medium frequency oscillation during the rest of the response (200us) can be seen. A zoom in on the high frequency oscillation shows a period of around 500ns modulated by a lower frequency signal.
- Figure 3d illustrates the disappearance of the peak and the over-damping response in 200us.

Equation 1 presents the equation to estimate the magnetic field:

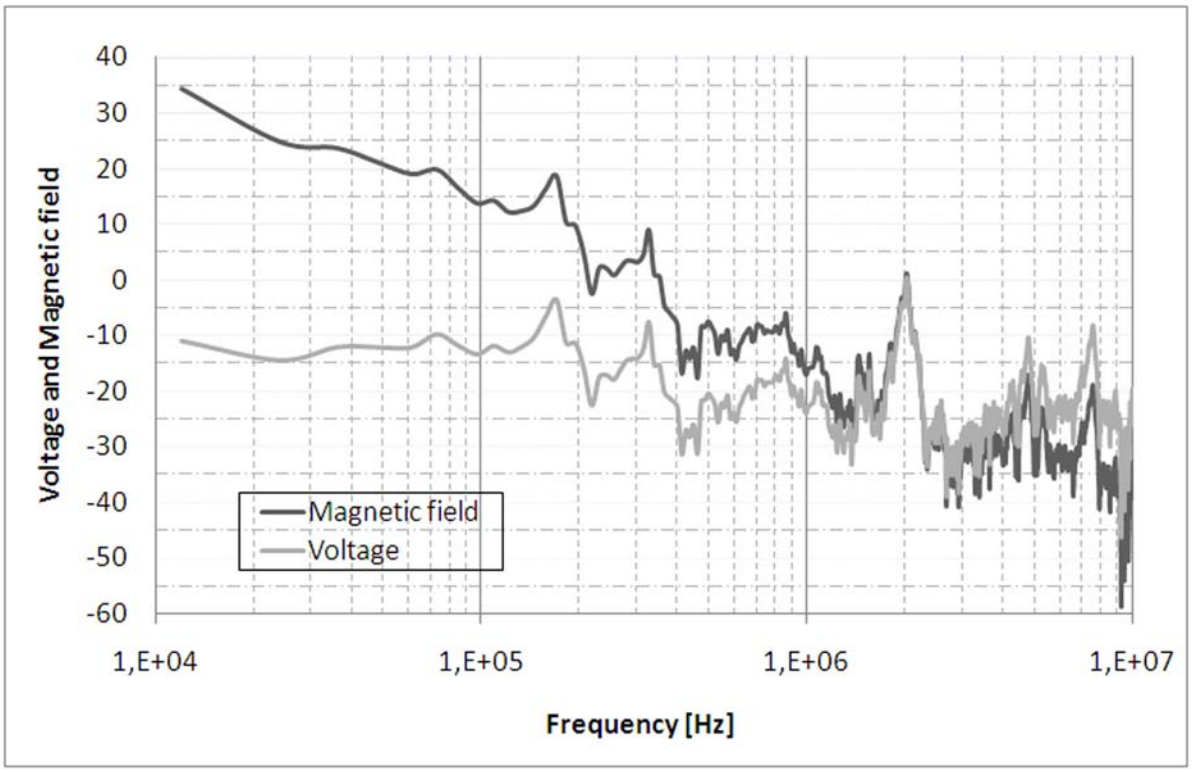
$$E = B \cdot S \cdot N \cdot 2 \cdot \pi \cdot f \quad (1)$$

where B is the magnetic field, E is the induced voltage which can be calculated from the oscilloscope measurement and applying the gain of the circuit between the antenna and the oscilloscope, S accounts for the area of the antenna (10cm x 11cm), N is the number of turns of the loop (5), and f stands for the frequency.

The normalized magnetic field intensity spectrum was calculated by means of an FFT applied to the time varying signals, and the results are shown in Figure 4.

The spectrum shows that the most important peaks, around 165kHz and 330kHz, were located around the operating frequency of the ASFA system and could cause its malfunction. Regarding BTM operating frequencies, a tone around 4.5MHz represents a source of interference. Another very large tone located at 2MHz was not close enough to the BTM operation frequencies. There were tones at frequencies above 7MHz that are higher than BTM oper-

Figure 4. Normalized magnetic field and voltage spectrum



ating frequencies and lower than LTM bandwidth, but they could be an interference for any other signalling located in these frequencies.

The noise spectrum for different signalling systems can be obtained using the test set-up defined in this section once the centre frequency and the bandwidth is identified. The obtained spectrum establishes the starting point for the definition of the reliability requirement that would assure the interoperability of the systems functioning in the rolling stock in its environment. The example presented in the third section is a spot signalling system.

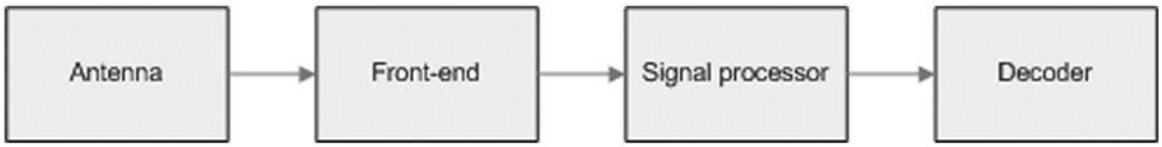
RELIABILITY REQUIREMENTS ASSESSMENT

The reliability of the communication is measured by means of the Bit Error Rate (BER), which is determined by the signal to noise ratio (*SNR*) at the receiver, modulation scheme used in the communication and the receiver performance. The higher the signal power received and the lower the noise at the receiver, the higher the communication reliability.

The received signal power depends on the following characteristics:

- The transmitter signal frequency and power,
- The distance between transmitter and receiver
- The existence of any obstacle between transmitter and receiver

Figure 5. Receiver block diagram



Maximum transmitted signal power is limited by the communication standard; therefore, in order to maximize SNR, noise has to be minimized. Electromagnetic noise found in the track is generated by a number of sources as defined in this chapter, however, electromagnetic railway environment noise emission where the communication systems are installed are not limited yet by a standard. Therefore, communication reliability between the rolling stock on-board equipment and the rail trackside equipment is affected by the electromagnetic environment found in the track.

A receiver is normally composed of an antenna, a front-end, a signal processor and a decoder (Figure 5). Its performance is determined by its design.

- **Antenna:** The receiver antenna acquires the magnetic flux or the electromagnetic signal sent by the transmitter antenna and converts it to an electrical signal. The antenna also performs the first filtering of the incoming signal at the bandwidth of interest. A passive antenna converts the signal and noise proportionally at the same ratio. On the other hand, an active antenna decreases the *SNR* of the incoming signal by increasing the thermal noise level by the active antenna's NF.
- **Front-End:** The front-end usually filters, amplifies and down-converts the incoming signal if necessary. As it is the case of an active antenna, a front-end decreases the *SNR* of the incoming signal by increasing the thermal noise level by the front-end´s NF. Additionally, digitalization is usually included in this block. In order to establish ADC characteristics, dynamic range, signal level and frequency of the incoming signal at the input of the digitalization has to be taken into account. Digitalising the signal decreases the output *SNR* by means of the quantification signal to noise ratio SNR_Q. Moreover, in order to avoid aliasing in the digitalized signal, proper filtering has to be included in the front-end.
- **Signal Processor:** Digital signal processing performs additional digital filtering to the signal and demodulates the signal for its further decoding. Signal demodulation quality is determined by the Bit Error Rate (BER) which is related to the energy per bit to noise power spectral density ratio (E_b/N_0).
- **Decoder:** The decoder is the responsible for obtaining the information messages from the demodulated signal, by means of the decoding process. It has not any effect on the *SNR* of the communication signal.

Communications reliability is defined by the BER at which the information is received. BER is related to *SNR* by means of the energy per bit to noise power spectral density ratio (E_b/N_0). E_b/N_0 can be defined as the *SNR* per bit, since it is a normalized measure of the *SNR*. It allows the BER of different digital modulation schemes to be obtained without taking bandwidth into account. E_b and N_0 are both expressed in watts per hertz and therefore E_b/N_0 is non-dimensional. The relationship between E_b/N_0 and SNR is defined by the following equation (Pearce 2000):

Table 1. Coherent and non-coherent detection comparison

Coherent detection	Non-coherent detection
– Require expensive and complex carrier recovery circuit – Better bit error rate of detection	– Do not require expensive and complex carrier recovery circuit – Poorer bit error rate of detection

$$SNR = \frac{R_b E_b}{BWN_0} \tag{2}$$

where R_b is the bit rate in bits/second, E_b is the energy per bit in Joules/bit and BW is the channel bandwidth in hertz.

It could be concluded that SNR will be more than E_b/N_0 by a factor of R_b, therefore when the data rate is increased, also the SNR would be. However, SNR is not increased by simply increasing R_b, since noise is also increased due to intersymbol interference. A trade-off between the data rate and the amount of noise that the receiver can handle has to be found.

Two modulation schemes of the ERMTS communications with Eurobalises and Euroloops are selected as examples to show the way of obtaining the BER of a communication by means of the E_b/N_0. This approach can be applied to any other communication system. First the modulation scheme is described, then the BER of the detection is plotted vs. the SNR of the input signal. There are two types of detectors (Table 1). A coherent detector uses the knowledge of the phase of the carrier to demodulate the signal, while if there is not phase information the only detection way is the non-coherent one.

Binary Phase Shift Keying (BPSK)

Loop Transmission Module (LTM) is the part of the ETCS on-board equipment that communicates with the Euroloops on the track. LTM and Euroloop are specified in (SUBSET044, 2004), where a Binary Phase Shift Keying (BPSK) modulation is defined for the communication between them. (SUBSET044, 2004), defines a BER of 10^{-4} after demodulation in the on-board equipment without any interference.

BPSK uses two phases which are separated by 180° to determine the logical '0' and '1'. High data-rate applications, when bandwidth is limited, do not employ this modulation since it is only able to modulate at 1 bit/symbol. However, it is the most robust of all the PSK modulations since it takes the highest level of noise or distortion to make the demodulator reach an incorrect decision and therefore suitable for railway environments.

The general form for BPSK follows the equation:

$$s_b(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi(1-n)), n = 0,1 \tag{3}$$

where f_c is the frequency of the carrier-wave and T_b is the bit duration.

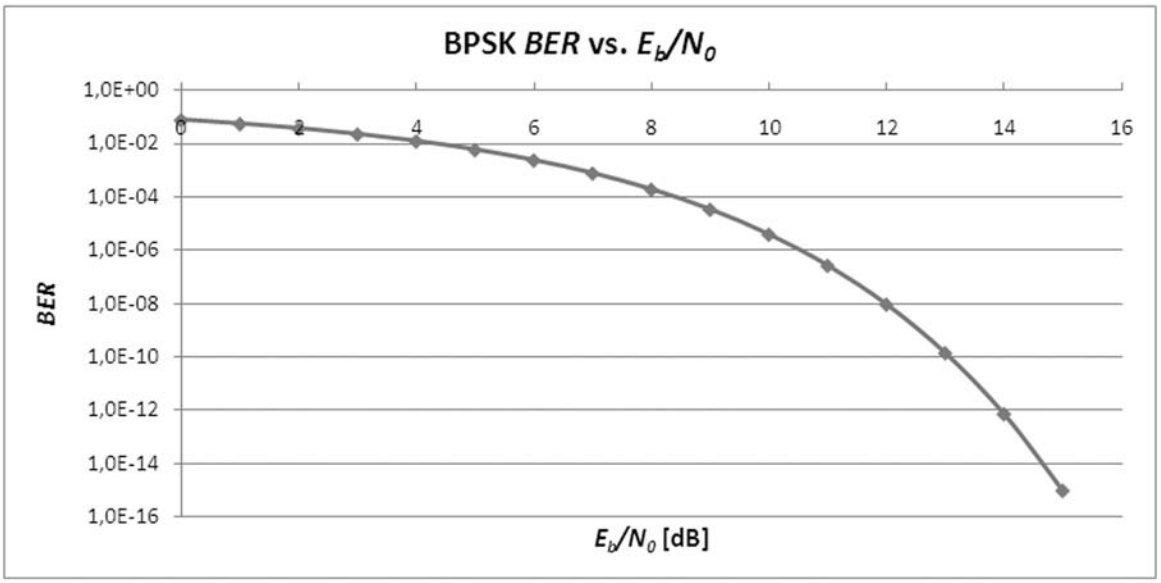
Binary data is often conveyed with the following signals with the two phases, 0 and π.

'0'

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t + \pi) = -\sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \tag{4}$$

'1' $s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t)$ (5)

Figure 6. BPSK BER vs. E_b/N_0



BPSK BER depends on E_b/N_0 and can be calculated by Equation 6 and is plotted in Fig. 6. Since BPSK shows one bit per symbol, BER is also the symbol error rate (Glover 2003).

$$BER = \frac{1}{2} erfc\left(\sqrt{\frac{E_b}{N_0}}\right) \tag{6}$$

Binary Frequency Shift Keying (BFSK)

Balise Transmission Module (BTM) is the part of the ETCS on-board equipment that communicates with the Eurobalises on the track. The BTM and the Eurobalise are specified in (SUBSET036, 2005), where a Continuous Phase Binary Frequency Shift Keying (CPFSK) modulation is defined for the communication between them. (SUBSET036, 2005) defines a BER, less than 10^{-6}, for both Up-link and Down-link in the central area of the contact length of each Eurobalise.

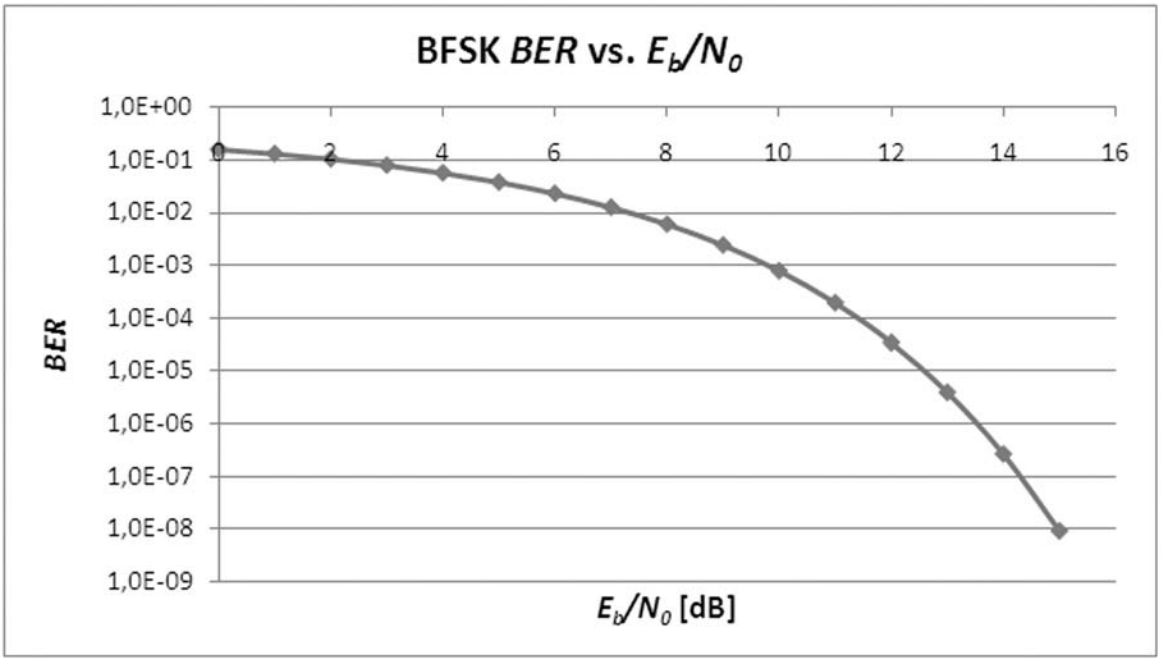
CPFSK is a modulation scheme based on the Binary Frequency Shift Keying (BFSK), where the phase of the signal is continuous. BFSK uses a pair of discrete frequencies to determine the logical '0' and '1'. This is the simplest FSK modulation, where digital information is transmitted through discrete frequency changes of a carrier wave. It is the most robust of all the FSKs, and therefore suitable for railway environments.

BFSK binary data is obtained by means of the signals describe by the equations:

$$\text{'0'}\ s_0(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_0 t + \theta_c), 0 < t \leq T \tag{7}$$

$$\text{'1'}\ s_1(t) = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_1 t + \theta_c), 0 < t \leq T \tag{8}$$

Figure 7. BFSK BER vs. E_b/N_0



where f_0 are f_1 are the frequencies of the carrier-wave for the logical '0' and '1' respectively and θ_c is the phase of the signal.

BFSK BER depends on E_b/N_0 and can be calculated by Equation 9 and is plotted in Figure 7. Since BFSK shows one bit per symbol, BER is also the symbol error rate [Glover 2003).

$$BER = \frac{1}{2} erfc\left(\sqrt{\frac{1}{2}\frac{E_b}{N_0}}\right) \quad (9)$$

In summary, in order to obtain the communication reliability of a signalling system, first the noise level in the environment on the working frequency band has to be obtained. This depends on where the antenna is located, which is consequently a key point in the measurement process. Then, by means of the minimum signal power and the receiver performance the available *SNR* at the demodulator has to be calculated. By means of the Equation 2, the communication E_b/N_0 can be obtained, and depending on the demodulation scheme, the BER of the communication can be achieved for the given E_b/N_0 as it has been shown for the BPSK and BFSK modulation scheme examples in Figure 6 and Figure 7.

CONCLUSION

The reliability assessment of a signalling system can be determined by the railway electromagnetic environment. The relationship between the EM interferences and the reliability requirements presented in this chapter improves the reliability analysis of any signalling system. The first two points of this chapter define these interferences in the railway environment and the description of the test set-up to characterize them. These are the key points to obtain the noise spectrum that limits the communications and fixes the reliability requirement that are necessary for the reliability assessment also defined in this chapter. The FP7

funded TREND project will continue with the definition of the railway electromagnetic spectrum characterizing the noise in the frequency bands of signalling systems in order to improve the reliability analysis of these systems.

ACKNOWLEDGMENT

The research Project TREND has received funding from the European Community's Framework Programme FP7/2007–2013 under grant agreement nº 285259. Consortium: CEIT (E), CAF group (E), CEDEX (E), IFSTTAR (F), YORK EMC Services (UK), LTU (S) and Trafikverket (S).

REFERENCES

del Portillo, J., Osinalde, M., Sukia, E., Sancho, I., Medizabal, J., & Meléndez, J. (2008): Characterization of the EM environment of railway spot communication systems. In *Symposium on Electromagnetic Compatibility, 2008. EMC 2008*. IEEE International.

EN50121. (2006). *Railway applications - Electromagnetic compatibility – Parts 1, 2, 3-1, 3-2, 4 and 5*. CENELEC.

EN50215. (1999). *Railway applications. Testing of rolling stock after completion of construction and before entry into service*. CENELEC.

EN50238. (2003). *Railway applications, Compatibility between rolling stock and train detection systems*. CENELEC.

EN50338. (2001). *Railway applications. Power supply and rolling stock. Technical criteria for the coordination between power supply (substation) and rolling stock to achieve interoperability*. CENELEC.

ERA EMC Report. (2010). *67575_ERA_EMC_Final_Report - Study to collect and document rules, processes and procedures to verify the electromagnetic compatibility of railway vehicles in member states of the European rail area, for ERA*. Lloyd's Register Group.

Glover, I., & Grant, P. (2003) *Digital Communications (2nd ed)*ISBN978-0130893994, Upper Saddle River, NJ: Prentice Hall

NERA. (2000). *Safety Regulations and Standards for European Railways*. NERA.

Niska, S. (2008) *Measurements and analysis of electromagnetic interferences in the Swedish railway systems*. Doctoral thesis Luleå tekniska universitet

PD CLC/TR 50507(2007) *Railway applications. Interference limits of existing track circuits used on European railways*. Dolecek,R. & Hlava, K. (2007): Transient Effects at Power-Supply System of the Czech Railways from EMC Viewpoint.. *RADIOENGINEERING16*(1)

Pearce, J. (2000) What's All This Eb/No Stuff, Anyway?, *Spread Spectrum Scene Online 7*(1)

RAILCOM (2008): *Granted European Project: Electromagnetic compatibility between rolling stock and rail-infrastructure encouraging European interoperability*. Cordis FP6

Slimen, N., & Deniau, V. (2008): On Board Measurements of the Railway's Electromagnetic Noise with Moving Train. In Slimen, N., Deniau, V., Baranowski, S., Rioult, J., Dubalen, N., & Démoulin, B. (eds.) *Consortium Railcom. Proceedings, 18th Int. Zurich Symposium on EMC*, Munich

Std, I. E. E. E. 610.12-1990 (1990) *IEEE Standard Glossary of Software Engineering Terminology* Washington, DC: IEEE Press

SUBSET036. (2005). *FFFIS for Eurobalise SUBSET-036. (2.3.2)*. UNISIG.

SUBSET044. (2004). *FFFIS for Euroloop SUBSET-044 (2.3.0.)*. UNISIG.

Tellini, B., Schneider, M., Petri, A., & Ciolini, R. (2008) Measurements of EM Emission in Rail Launcher Operation. *I2MTC 2008 - IEEE International Instrumentation and Measurement Technology Conference* Washington, DC: IEEE Press

TREND. (2007): *Test of Rolling Stock Electromagnetic Compatibility for cross-Domain Interoperability*. Research Project funded by the European Community's Framework Programme FP7/2007–2013 under grant agreement nº 285259. Consortium: CEIT (E), CAF group (E), CEDEX (E), IFSTTAR (F), YORK EMC Services (UK), LTU (SE) and Trafikverket (SE).

Vromans, M. J. C. M. (2005). *Reliability of Railway Systems*. The Netherlands: TRAIL Research School.

York (2002): *Potential electromagnetic interferences to radio services from railways, final report for Radiocommunications Agency AY 4110*. York EMC Services LTD.

York (2004): *Improved methods for the measurement of radiofrequency emissions from railways AY 4365*. York EMC Services LTD.

KEY TERMS AND DEFINITIONS

ADC: Analogue to Digital Converter

AM: Amplitude Modulation

ASFA: Anuncio de Señales y Frenado Automático

BER: Bit Error Rate

BFSK: Binary Frequency Shift Keying

BPSK: Binary Phase Shift Keying

BTM: Balise Transmission System

CENELEC: European Committee for Electrotechnical Standardization

DAB: Digital Audio Broadcasting

EM: Electromagnetic

EMC: Electromagnetic Compatibility

EMF: Electromagnetic Fields

EMI: Electromagnetic Interference

ERA: European Railway Agency

ERTMS: European Railway Traffic Management System

ETS: European Telecommunications Standards Institute

EU: European Union

FFT: Fast Fourier Transform

FM: Frequency Modulation

FP: Framework Program

FSK: Frequency Shift Keying

GSM-R: Global System for Mobile Communications - Railway

IEEE: Institute of Electrical and Electronics Engineers

IT: Information Technology

LTM: Loop Transmission Module

LZB: Linien Zug Beeinflussung

NF: Noise Figure

PSK: Phase Shift Keying

SNR: Signal to Noise ratio

TEN-T Railway Network: Trans-European Network for Transport

TSI: Technical Specifications for Interoperability

Section 4

Automation in Development and Testing

Chapter 8
Mívθα:
A Framework for Auto-Programming and Testing of Railway Controllers for Varying Clients

Jörn Guy Süß
University of Queensland, Australia

David Carrington
University of Queensland, Australia

Neil Robinson
RGB Assurance, Australia

Paul Strooper
University of Queensland, Australia

ABSTRACT

Implementation of railway controller application logic is a highly safety-critical and time-consuming task carried out individually for each client and station by specialised signalling engineers, with corresponding high costs. Mívθα is a software development framework designed to create code generators for application logic for the client railway companies of Ansaldo STS that use the Microlok II controller to lower the cost and increase repeatability. This chapter describes the evolution of Mívθα from prototype to framework, and introduces the software engineering approaches of object-oriented meta-modelling and framework development along the way. It also presents known limitations and further application areas of the framework.

INTRODUCTION

Many advances in the development of rail technology have been accompanied by an increase in risk to the passenger. Shared track use meant an increased number of trains, but carried the risk of rear collisions; points brought an increase in route flexibility, but at the cost of potential derailment and head-on collisions; higher train speeds allow passengers to travel faster than small planes, and metro rail trains accelerate and decelerate in shortest distances, but these types of trains cannot be driven by sight alone any more. Each increase in risk has been managed by signalling engineers through the introduction of corresponding increasingly sophisticated control machinery. The safe conduct of today's rail traffic relies on computer-based interlocking controllers. The programs that these controllers execute reflect the whole complexity scope of signal engineering.

DOI: 10.4018/978-1-4666-1643-1.ch008

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Since interlocking controllers are specialised computer devices, the creation of application logic for these controllers is essentially a software-programming task. The professional discipline that examines and develops methods and tools for the production of software is software engineering. The Mint project is a software engineering project that investigates the process of writing interlocking controller logic to suggest ways to improve the process. The Mint project applies three software engineering techniques to deal with the complexity of the domain: object-oriented modelling to capture and describe the artefacts and terminology of signal engineering, feature modelling to describe the varying aspects among different clients, and framework development to reduce the cost of construction by providing a common infrastructure.

This chapter introduces the software engineering approach of the Mint project in detail to show how it derives client-specific tools that automate the process of generating controller application logic as far as safely possible.

BACKGROUND

The Mint project is a joint project of Ansaldo STS and The University of Queensland. In the following section, we introduce Ansaldo and its business requirements. After that, we turn to Ansaldo's Microlok system that forms the target platform of the Mint project, and describe its functionality in detail.

Ansaldo STS – International Railway Engineering

Ansaldo STS is a global railway engineering service provider that specializes in automation and turnkey delivery of passenger and freight rail systems. Ansaldo STS was created through a merger between Union Switch and Signal and Ansaldo Trasporti Sistemi Ferroviari, and US-American and European technologies determine its offerings. In 2009, Ansaldo STS achieved revenue of €1.176 billion and an operating income of €125.0 million, with 4,340 employees worldwide. Ansaldo's clients vary considerably by their operational objectives and implementation of railway principles. Clients range from metropolitan rail operators to mining rails, and stock spans from electrified high-speed locos to double-traction diesels. Communication may be by coded track, radio, or GSM-R; signals may be physical installations at trackside, or console indicators in the driver's cabin. Ansaldo's task is to manage these variations to provide safe signalling systems that respect the clients' existing customs and requirements.

The Microlok System

One of Ansaldo's most successful products is the system of Microlok integrated controllers, designed by Union Switch and Signal. Before the arrival of the Microlok system, signalling controllers were usually custom-built devices that were programmed in machine language. The Microlok I controller introduced a language that signalling engineers found easy to adopt, because it modelled the wiring logic of the relay grids that they were familiar with. The Microlok II controller expanded the approach by applying the principle of the IBM personal computer to the signalling system: a Microlok system consists of a powered case that holds a central-processing unit and several interface cards on separate plug-in boards that connect to the CPU via a common bus. While the Microlok architecture was designed to serve as a platform for vital controllers, the standardisation also allowed Ansaldo to build a non-vital controller as a parallel product to share the components.

The CPU of the Microlok II is built on the well-established and understood Motorola 68000 series of microprocessors. The bus between CPU and cards follows the VME Bus standard (IEEE, 1987), which enables the use of standardised electronics testing and design facilities.

Figure 1 shows the information flow of the Microlok II controller. Inputs (on the left) report

the state of the interlocking. Inputs include regular track circuits, audio frequency tuned circuits, detection rods, keys, buttons, requests from the Train Control System (TCS), and proofing information from adjacent stations for the overlap. The Microlok connects these inputs using application logic and produces outputs to actuators, such as signal lamps, switch machines, and boom lifts. It also indicates actions and conditions of the interlocking to the TCS. Some computations require a delayed response. For example, a signal may delay showing a 'proceed' aspect until a train has been on its approach track circuit long enough to be sure the train has stopped. To cover this requirement, the Microlok provides timers. To protect the integrity of the system, the Microlok also includes several diagnostic inputs and outputs that represent the health of the controller and its plug-in cards and deal with implications for the whole installation. For example, installations usually contain a Vital Cut-Off Relay (VCOR), which is controlled by the system's vital CPU board logic. If a failure of the controller's hardware is detected, VCOR is dropped and all vital field equipment returns to its most restrictive state, making the interlocking safe.

Internally, the Microlok derives the state of the output via several intermediate steps. Figure 2 shows an example of Microlok II application logic that controls a points-machine. The application has to fulfil the following requirements:

- If the Train Control Centre requests that the points be moved, and the points are not in the desired position, the points-machine should run.
- The request must be abandoned if the machine is already running, or a train is currently detected. The move should either be completed first, or prevented.
- The request must also be abandoned if the points-machine is jammed. The machine is jammed if it takes longer than 20 seconds to move the points to the end position.

In the program fragment below the diagram, the asterisk denotes a Boolean 'and' operator, the tilde denotes 'not', and the plus denotes 'or'. In the first statement, the clause in parenthesis defines the start condition: if the centre asks for the points to be in the normal position, (**N**LR) and they are not there (**N**WKR), or analogous for the reverse position, the motor will start. This is not the case if a train is detected (TP) or the motor is already running (WCR). If the motor is jammed (WJR), it should not run. Normally, the second clause would immediately set the WJR variable to true. However, WJR is linked to a twenty-second timer that delays its change. WCR is also delayed. This ensures that an intermittent failure to detect a train, for example due to dirty or iced-over rail heads or bouncing axles, does not permit the motor to start unexpectedly.

The historical use of relays has led to a tradition of terse mnemonic identifiers for the names of variables used in the application logic. Table 1 shows the mnemonics used in the example.

CHALLENGES OF APPLICATION LOGIC DEVELOPMENT

Three major issues make the development of Microlok application logic difficult:

- Tests of the software require installation on controller hardware. This leads to long turn-around times between software tests.
- The procedures used to derive application logic code from track layouts and control tables are given by example, they are often incomplete and may change during the course of the project.
- The way in which control tables, track layouts and procedures for development are defined varies for every client, making reuse challenging.

Figure 1. Microlok II Information Flow

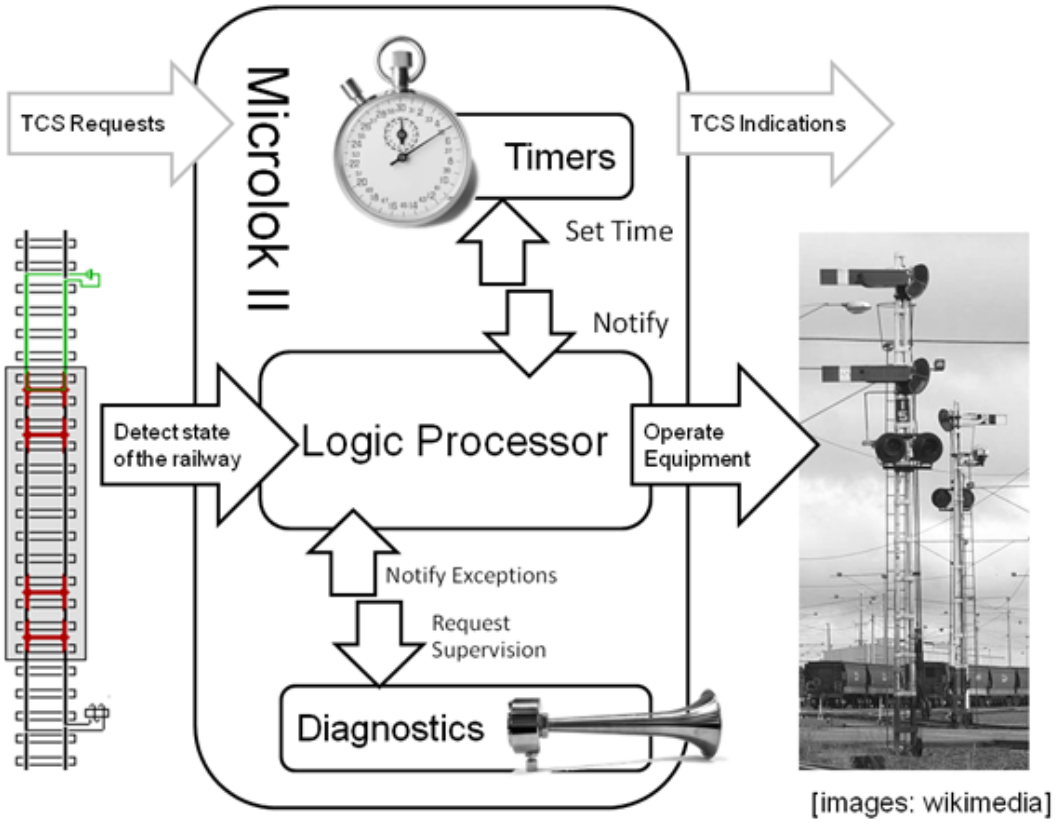
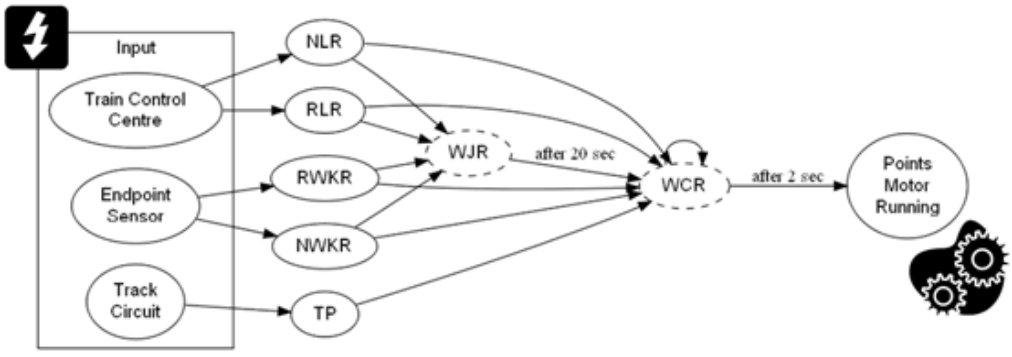


Figure 2. Application Logic Example for a Points Control Operation



// Start the motor ➔ Raise after 2 seconds, drop immediately
ASSIGN (NLR * ~NWKR + RLR * ~RWKR) * ~(TP + WCR) * ~WJR **TO** WCR;

// Check for motor jam ➔ Raise after 20 seconds, drop immediately
ASSIGN NLR * ~NWKR + RLR * ~RWKR **TO** WJR;

Table 1. Relay naming conventions

Code	Meaning
W	Switch – from the air points look like a 'W'
N	Normal – Trains pass straight over the switch
R	Reverse – Trains are diverted to the other line
L	Call (from train control system)
C	Command (to the points machine) – "Move the points"
K	"OK" – Operation Complete, points detected in desired position
J	The points motor has run too long without successfully moving the points (a possible "Jam")
TP	Track Proven (to be occupied) – There is a train on this track

The following sections describe the current process and discuss how the issues arise.

Turnaround Times of the Development Process

Figure 3 shows a schema of the process that Ansaldo uses to develop application logic. Downward arrows indicate that code is being passed to the next worker; upward arrows indicate it is passed back because it is faulty. The width of the upward arrows indicates the frequency of faults. The width of the downward arrows indicates the size of the artefact that is being passed on.

The process begins with the production of the application logic code by a functional code developer. The code is subsequently read by a second developer to verify its correctness. Afterwards a simulation engineer configures a computer model of the actual station in the Microlok System Simulator (MISS) software. The application logic code is installed on a Microlok rack, and the rack is connected to MISS. A principles tester then uses this simulation of the complete station to test the application logic for issues. MISS does not generate test scenarios. The principles tester has to place and move trains in the interlocking to detect faults. Most faults are discovered during this phase of development.

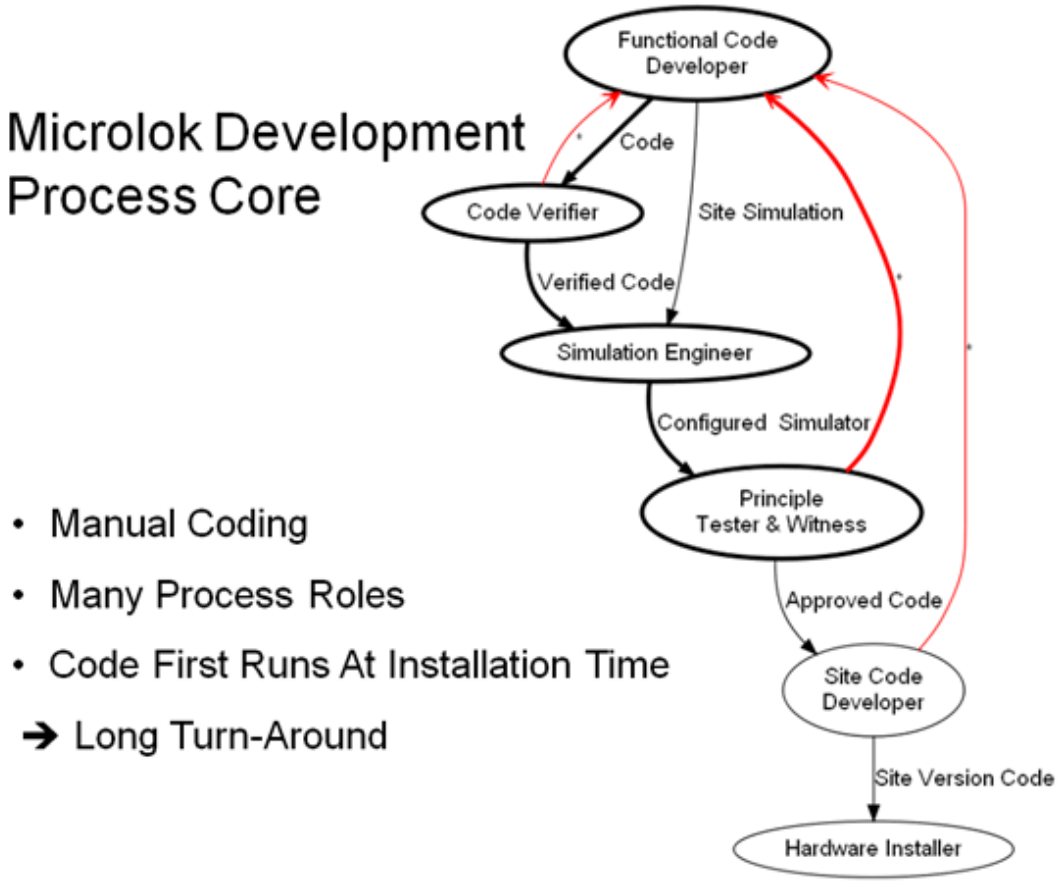
The length of the cycle from development to the first test represents a challenge to the development process. In principle, every change to the original code will require that all tests of the station be repeated. However, a complete repetition is unfeasible due to the high cost of the manual testing process. Instead, the testers assess the expected change impact based on their experience and selectively create scenarios to evoke corresponding faults. The testers keep a signed record of the safety of the various functional aspects of the interlocking, but they do not record the scenarios that have actually been tested. This can make it difficult to analyse if a fault that is observed in the field at the actual installation site is due to differences between the simulation and actual site, or if it has been overlooked in the testing phase.

Changeable and Incomplete Specification

The example in Figure 2 does not describe which points are being controlled. Instead, the example forms a pattern that can be used to create control programs for any point, by prefixing the variable names with the number of the point and associated track. For example, NLR of point 13 would be 13NLR and track detection for track 2 would be 2TP. The creation of general patterns forms an important part of Ansaldo's development cycle.

The patterns form a catalogue of approved solutions. This catalogue is part of the Application

Figure 3. Application Logic development process

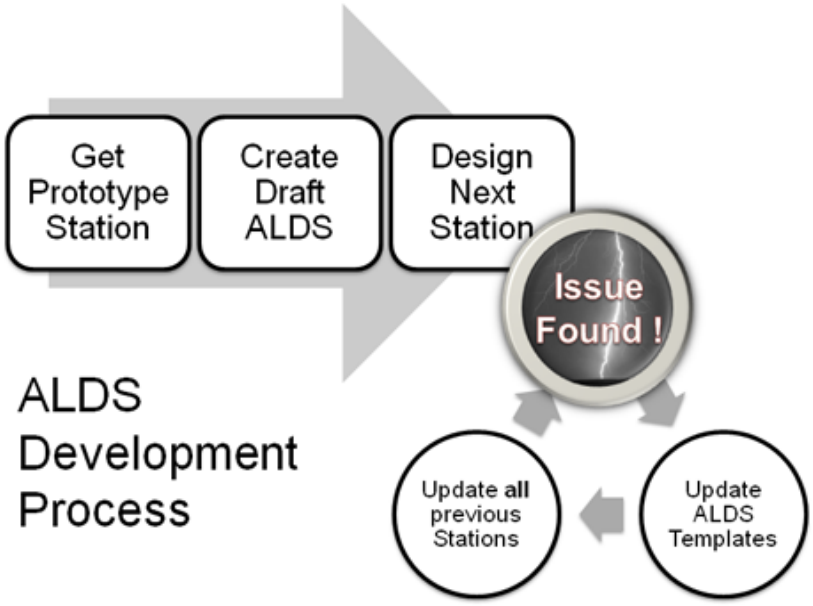


Logic Design Specification (ALDS). The ALDS is the document that directs the development of application logic for a specific client and specifies constraints on the outcomes. Figure 4 describes how the ALDS is developed. First, a station that is perceived as a typical representative of all stations in the project is analysed, and all patterns are extracted to form the draft ALDS. Afterwards, all other stations are derived from the patterns laid down in the ALDS. If errors or omissions are discovered during the derivation of a new station, the ALDS is reworked. If the changes affect previously designed stations, the stations have to be updated to the latest standard. Since all development is carried out by railway engineers without automation support, change management is often challenging and the cost of development is high. Our project aims to automate the creation of controller logic for individual stations from the ALDS, but to retain Ansaldo's current development process.

Client Variations

Each client uses different conventions for the description of control tables and track layouts. In addition, procedures of the companies also differ by client. Attempts for automation are limited by this variability. Automation is achievable, but it is too costly to develop for one client. Our project aims to create reusable components and a method of composition that allows the construction of a solution at a cost lower than that of individual development.

Figure 4. Lifecycle of a project and its ALDS



MODELLING SIGNAL ENGINEERING KNOW-HOW

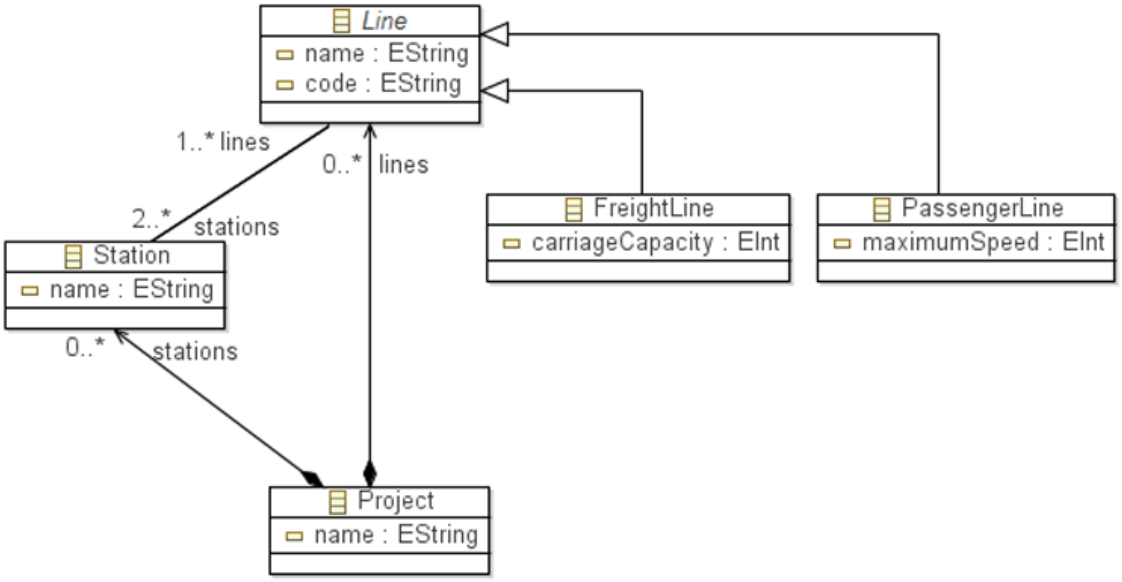
Engineering disciplines are characterised by the ability to describe the relevant issues of the objects of their domain. Engineers employ a defined set of terms, and can explain the relationships among those terms in an unambiguous form. In order to create computer tools for an engineer, the software must capture and encode the objects, their relevant properties, and their relationships. This involves an analysis of the engineering domain, followed by gradual development of the software by a programmer who interprets the results of the analysis and translates it into the program. Such a conventional development process for engineering software is slow, because it spans numerous phases, and error-prone, because the developer receives the description of the domain from a software analyst, and not from an engineer. For example, to create a software program for greenhouse climate management, a biologist would speak to an analyst, who would pass information to the developer.

Class Models

To reduce this overhead and foster the creation of computer software for areas where it would otherwise be infeasible, software engineering research has developed a technique that uses explicit language models as the basis for software development. Figure 5 shows a simple example of such an object-oriented model that describes the organisation of a railway project. The diagram shows five compartmentalised rectangles that represent *classes*. A class groups objects of the same kind within a domain. For example, 'Wotonga Duplication' and 'Southwest Mining' could both be Project objects, classified by the Project class. The top compartment shows the name of the class, the compartment underneath shows the corresponding properties. Project objects have a name, that is a String, and PassengerLine objects have a speed, that is an Integer number. The properties of a class are just values; they do not have any further internal structure.

Classes engage with each other via relationships, shown as lines. The relationships represent

Figure 5. A simple model of rail projects that demonstrates the object-oriented notation

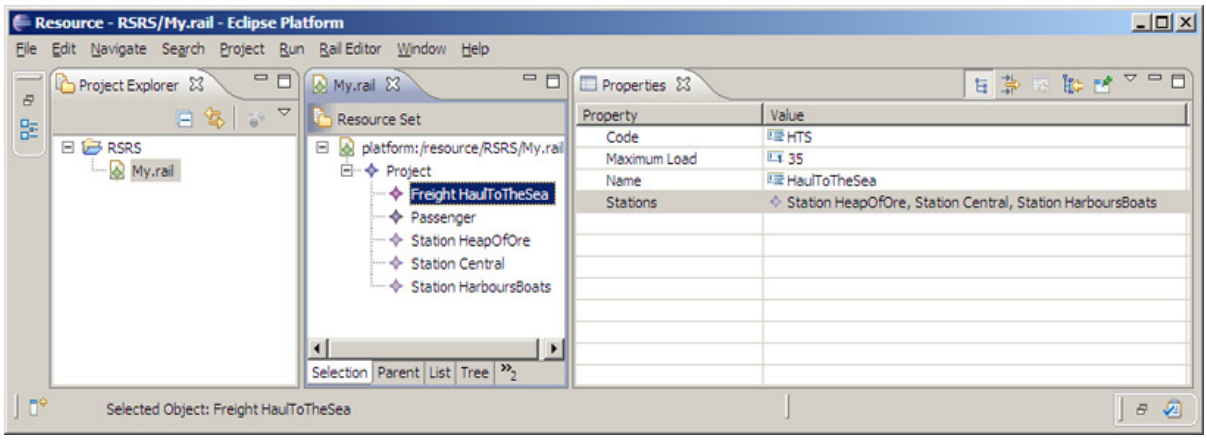


common concepts of engineering thought. A line that starts with a filled rhombus symbol indicates that the class that has the rhombus contains the class at the opposite side. In the example, a Project contains one or more Stations. This relationship is a composition. A simple line indicates that the connected classes are related. For example, Stations and Lines are related. This relationship is an association. To limit the number of associated items, the far end of an association can carry a Multiplicity marker. The asterisk (*) is a wildcard that indicates that any number is acceptable. For example, a minimum of two Stations must be associated with each Line. A hierarchy of generalisation among concepts appears in the diagram as a filled arrowhead that points at the abstract concept. For example, a FreightLine is a special type of line that can enter into all relationships the general line can enter into, and has all the properties of the regular Line in addition to those that are specific to it, like the carriageCapacity in this example. In some cases, the general class will be an abstract concept. In our example, we can create FreightLines and PassengerLines, but abstract Lines are not meaningful and hence prevented. To identify them, the name of an abstract class like Line is italicised.

The notation introduced above is a constrained version of class diagrams that form part of the Unified Modelling Language (UML) (OMG2005-umls, 2005). The class models are used as input to the Eclipse Modelling Framework (EMF) (Budinsky et al., 2003). Modellers can use this software to draw these class models of the engineering language. The constrained diagrams are only intended to represent the *core* concepts of EMF, and are hence called eCore models or simply eCores. To distinguish the *modelled* data types of EMF from the plain classes of the underlying Java programming language, the type names are prefixed with an 'e', so a modelled String is of type 'eString' and so forth.

EMF provides a translator to process the eCore models to produce a corresponding software tool that allows capturing any set of objects that conform to the requirements of the diagram. Figure 6 shows the tree-based editor of the rail project manager generated from the example diagram in Figure 5.

Figure 6. the editor program that was automatically generated of the example model



Mint in based on EMF and has been using the freely available Eclipse modelling tools edition of the platform that contains the editor and generator tools mentioned above, as well as some tools for model transformation, as described in the following section.

Model Transformation

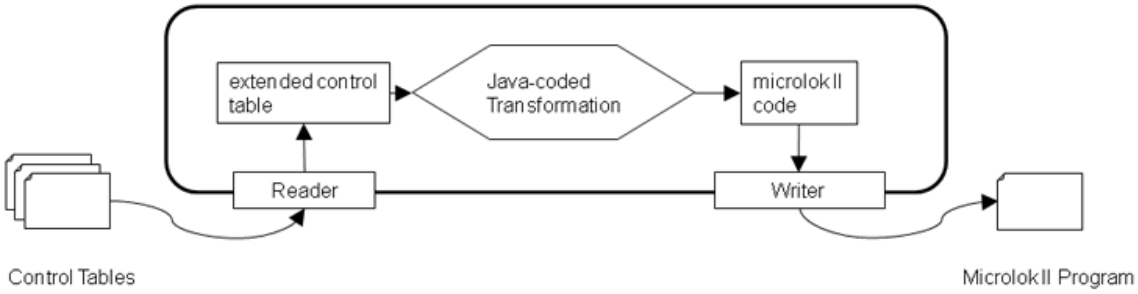
Most engineering domains will comprise too many aspects to fit into just one eCore model. Such a large model would be difficult to handle and even more difficult to understand. Consequently, model-based projects design separate eCore models for the different aspects of the engineering domain in question. These models necessarily share some common aspects, and so a program can compute the shared parts of one model from the other model, thus avoiding the work of re-entry or manual computation. If we were to create another eCore model to express the track plan of a station it might also contain information held in the Line and Station classes in the management model shown in Figure 5. To transfer information from one model to the next, or to compute the content of an output model based on one or more input models, model transformation languages can be used. These languages query the content of the input models, select the relevant parts, and project them into the output model.

Applications of Model-Driven Engineering

Over the last five years, industry has adopted the Model-Driven Engineering (MDE) approach (mil, 2001). Companies like BMW use it to describe automotive electronics; Saab Bofors Dynamics uses it to simulate the behaviour of missiles, and Motorola uses it to define electronics of network switches and mobile phones in a model-driven fashion (Baker et al., 2005). These companies' businesses, like that of Ansaldo, all require the programming of microcontrollers from higher-level domain descriptions. MDE is particularly effective for this application.

Within the railway signalling industry, there are a number of existing toolsets for generating application logic for interlockings. Most are specialised toolsets designed to support particular interlockings. For example, Ansaldo itself offers two other interlocking controller products, SEI and ACC, aimed at different kinds of client applications. Both these interlockings have toolsets that support the automatic generation of application logic. Westinghouse, Alstom, and Bombardier also have similar toolsets. Prover Technology AB specialises in providing toolsets for signalling applications and provides

Figure 7. QR Regional prototype architecture



customisable tools for design, proofing and testing – from our survey of such tools, the Prover Technology solution appears to represent the current state of the art.

The upfront costs of adapting these toolsets for a customer's application are very high, but these costs can be recovered over a number of subsequent applications for the same customer. This is appropriate in signalling markets in which there are standardised signalling rules and a volume of work for each customer that justifies the upfront costs, as is the situation in Europe. However, in some areas of the world, including in Australia, there are many different customers with relatively small railways and there is less standardisation in signalling rules. In such areas, the costs of adaptation using the kinds of toolsets described above are currently prohibitive. An MDE approach on the other hand offers the potential to perform adaptations for each customer at a relatively low cost.

A PROTOTYPE

The Mint project began as a case study in the application of model-driven technology. The Microlok controller had similar properties to the target systems of successful model-driven projects. It is a microcontroller whose program is developed based on rules laid down in the ALDS and information presented in a domain-specific model: a set of control tables.

From Control Tables to Application Logic

The resulting prototype was specific to Queensland Rail Regional (QRR) signalling requirements. It consisted of two models and one transformation, as shown in Figure 7. The first model represented the control tables, the second the Microlok II application logic. The program read input from a Microsoft Excel Worksheet, transformed it, and wrote the output Microlok program to a text file.

A Hand-Written Transformation

The model transformation used in the prototype did not use a general-purpose transformation language, but Java, a general-purpose programming language. The transformation implemented the patterns of the QRR ALDS, and produced an output section of code for each input element in the control table. The fragment of the ALDS previously introduced is actually part of the QRR ALDS. The control table models each piece of trackside equipment that the interlocking has to control, including signals and points. For each point listed in the table, the transformation produced a corresponding segment of code in the code model on the output side.

Limitations of the Prototype

As we added more ALDS patterns to produce a more complete output program, we discovered that certain rules required more information than was available in the control tables. For example, for the creation of locking conditions the signalling engineers referred to the track layout to identify incoming connections. To overcome this, the track layout needed to be added as an additional model. In addition, the control table contained annotations, known as hash marks. The meaning and intended effect of these hash marks varied substantially. One group served as abbreviations for repeated expressions, another altered the meaning of the column entries, and a third mandated the generation of a specific form of Microlok code in the output. We found that Queensland Rail had about 100 of these marks catalogued, but that this catalogue was not closed. In response to these issues, the prototype was refined by Ansaldo's software engineers to produce an invalid program statement intentionally whenever more information was needed, augmented with an explanatory comment to explain the requirement. The invalid statement prevented unintentional use of the automatically generated code and ensured that a signalling engineer would inspect and adapt the program as necessary.

Ansaldo was interested in the original prototype, produced a refined version in-house, and tested it. However, the high production time of six months made the software costly. At this point, Ansaldo and UQ initiated the Mint project to investigate how the approach could be accelerated and what other benefits, particularly in the testing area, could be provided. While Ansaldo repeated the development for other clients, using the development cycle and architecture of the prototype as a template, the research project at UQ looked at the creation of a method and set of reusable pieces to build a product line of code generators.

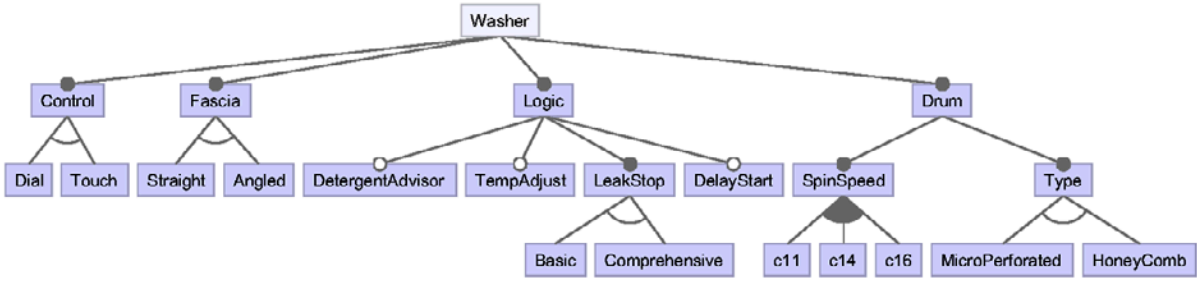
Managing Product Variability

Product line engineering is a management approach that optimises the utilisation of a company's capabilities. A product line is a set of related offerings created by configuring a set of variation points. For example, a washing machine company may produce a range of machines with varying drum-types, machine controls, displays, and an optional automated shut-off valve. In this scenario, the product is a washing machine, and the feature variations are in the drum, control, display, and valve. In principle, any combination of these features can be manufactured as a product. This combinatorial scope is only limited where features have interdependencies. In the case of the washing-machine example, a digital display may require computer control, and is incompatible with mechanical control. Marketing and customer relationship management can subsequently examine the market requirements and configure the products in the product line to suit the customer base.

Models of Product Lines

Product lines have been used in areas that involve mechatronic systems, where a hardware component is driven by an embedded controller running a built-in firmware (Weiss, 2008). For every product variation of the hardware, the corresponding firmware has to be different. Control systems for windshield wipers or fuel injection pumps are typical examples of this class of products. To produce the various driver programs without error, the programming task has to be automated. The design of space probes and spacecraft is a more exotic application of product lines. NASA is adopting this approach for the configuration of mission software (Pasetti, 2002).

Figure 8. Example of a Feature Model for Washing Machines



SOFTWARE PRODUCT LINES AND FRAMEWORKS

The idea of product lines can also be applied to software *without* any physical component (Eisenecker & Czarnecki, 2000). In the Mint project, Ansaldo's clients vary in the way they describe projects and stations, and translate the requirements into Microlok application logic.

In physical products, a number of product factors are fixed and cannot be changed. These elements constitute an invariant frame into which the varying components have to be installed. As a rule, the fixed parts substantially outnumber the varying parts. Along with the frame, there is usually a definition of defaults to be used in case no explicit variant is specified. For the example of the washing machine, this would mean that the outer metal housing of the machine might be a common frame into which the variant drum is installed. By default, a 'standard' type washer may include a punch-hole medium-size drum, and a mechanical motor control.

The idea of frameworks as a base for construction has also been adopted for the construction of software alongside product lines. In the Mint project, the parts of the tool that define the user interface, help system, and execution of the workflow are all part of a common software framework called Mívθα, which is filled in with the variant parts required by the clients.

To describe the variations in development frameworks, engineers use a modelling notation called feature models. Figure 8 shows an example of a feature diagram loosely modelled on the feature set of a German washing machine manufacturer. The boxes in the diagram are the varying features of the top node, the 'Washer' product. Features with a filled dot (●) on the top side are mandatory. Every product has to implement them. For example, no washer is complete without a drum. An empty dot (○) signifies that the feature is optional. For example, a washer would be usable without an automated detergent advisor. An empty arc or bow under a feature means that its sub-features represent a choice. For example, every washer needs a fascia, and it can be mounted with either a straight or angular mount piece. A filled bow indicates that a feature is mandatory, but potentially more than one feature is allowed. For example, each washer has to support at least one spin speed, but may offer more than one.

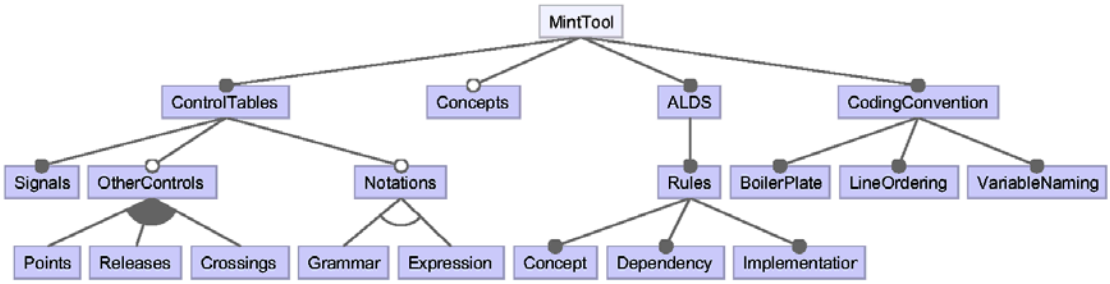
The diagram in Figure 8 has been prepared with FeatureIDE (Kastner et al., 2009), a free component for the Eclipse framework. The Mint project uses this component to model the space of solutions and to validate specific client instances.

In the following section, we will use the feature model notation to describe the design of the Mívθα framework.

Mívθα – A Framework for Modelling Interlockings

The Mint project applies the approach of software product lines to manage the heterogeneity

Figure 9. Feature Model of the Mint Tool



of Ansaldo's client companies with regard to their project definition information and ALDSs. A common framework provides the basis for the software product line. The name of the framework is Μίνθα, the Greek translation of Mint, to differentiate it from the industrial versions produced by Ansaldo and allude to its origin in academic research. The following sections describe the variations that Μίνθα manages in more detail, and how the framework is used to create an instance of the solution. Figure 9 shows the variation points of a Mint tool. We will refer to the diagram in the text below.

Client Variations

Ansaldo's clients vary in their business, according to the services they provide to their customers. Some carry passengers at high speeds from city to city; others haul iron ore over long distances in the Australian outback.

The different requirements naturally prompt different rolling stock and trackside equipment. Μίνθα contains a model that allows the developer of a Μίνθα-tool to specify what trackside equipment a client can install. These differences manifest themselves in the control tables, shown on the left side of the diagram in Figure 9.

Control tables describe how the interlocking controller should operate each piece of trackside equipment. All clients will use signals, so this feature must always be defined. The presence of other tables depends on the presence of the corresponding piece of hardware. The model in Figure 9 provides for clients to use any or all of the following devices: (motor-driven) points, (manual) releases, or crossings.

The entries within the columns of the equipment control tables are rarely just plain data, such as numbers or text strings. More often, they are short notations that express conditions or instructions. Signalling engineers design these notations to keep the tables concise. Using notations, the conditions that need to be specified for a specific type of trackside equipment can fit on a single large sheet of paper, rather than being spread out over several pages. If the client uses notations, these notations first have to be defined in the tool. Μίνθα incorporates two kinds of notations for different purposes: expressions and grammars.

Expressions are simple formatting templates that do not contain nested elements. For example, track identifiers may be constructed in the following way:

1. Uppercase Initials of Station Name
2. Track Number
3. The letter 'T'

Consequently, track 22 of 'Example Junction' could be denoted by 'EJ22T' in the control table. The strings '22T', 'E22T,' or 'EJ22'should be rejected by the framework. Expressions can be

defined by signalling engineers using a simple editor. In technical terms, expressions use regular expressions, a standardised notation for selecting parts of a text string.

Grammars are a more powerful mechanism that allows the definition of structures that include nested constructs. For example, a lock condition can have the form '17BT (27AT w 12R)'. Grammars need to be designed by a modeller. Technically, the parts of the framework that later recognise grammars are produced by a parser generator, a software engineering tool used to produces analysers for complex languages that allow nesting (Gagnon, 1998).

Figure 1 schematically shows the track status input and control output of the Microlok II in an interlocking. The relation between inputs and outputs is rarely direct. Usually it involves a computation with numerous intermediate steps. In the examples we analysed for the project, hierarchies of eight levels of evaluation were the norm.

To describe the items of the computation, a signalling engineer has a vocabulary of concepts. For example, there are a number of defined locks and locking strategies like Route locking, Sectional route locking, Approach locking, and Check locking. Signalling engineers understand the operational meaning of each of those defined concepts and can describe how they are derived in terms of the original input. For each of the locks mentioned in the example, a signalling engineer can name the conditions required to establish such a lock in a specific track layout of a station.

To separate the description of the signal-engineering matter from their implementation in Microlok II application logic, Mívθα first translates the control tables entered as text into representations of the signal-engineering concepts. In the second step, Mívθα transforms the concepts into code using ALDS rules.

The translation of the requirements into Microlok code varies with the ALDS that is used. Mívθα contains a model that describes the rules of the ALDS. A rule defines the concepts that it is based on, and the concepts it produces. Each rule is implemented by a programmer as a small program fragment. Where one rule requires another rule to be executed as a prerequisite, the framework infers the dependency and backtracks as necessary.

The order of lines of code and the names of the variables used in the code do not influence the behaviour of the Microlok program, but signalling engineers often refer to the source code listing of a Microlok application logic program as a specification document. This secondary use of the source code makes coding conventions important for the Mint project.

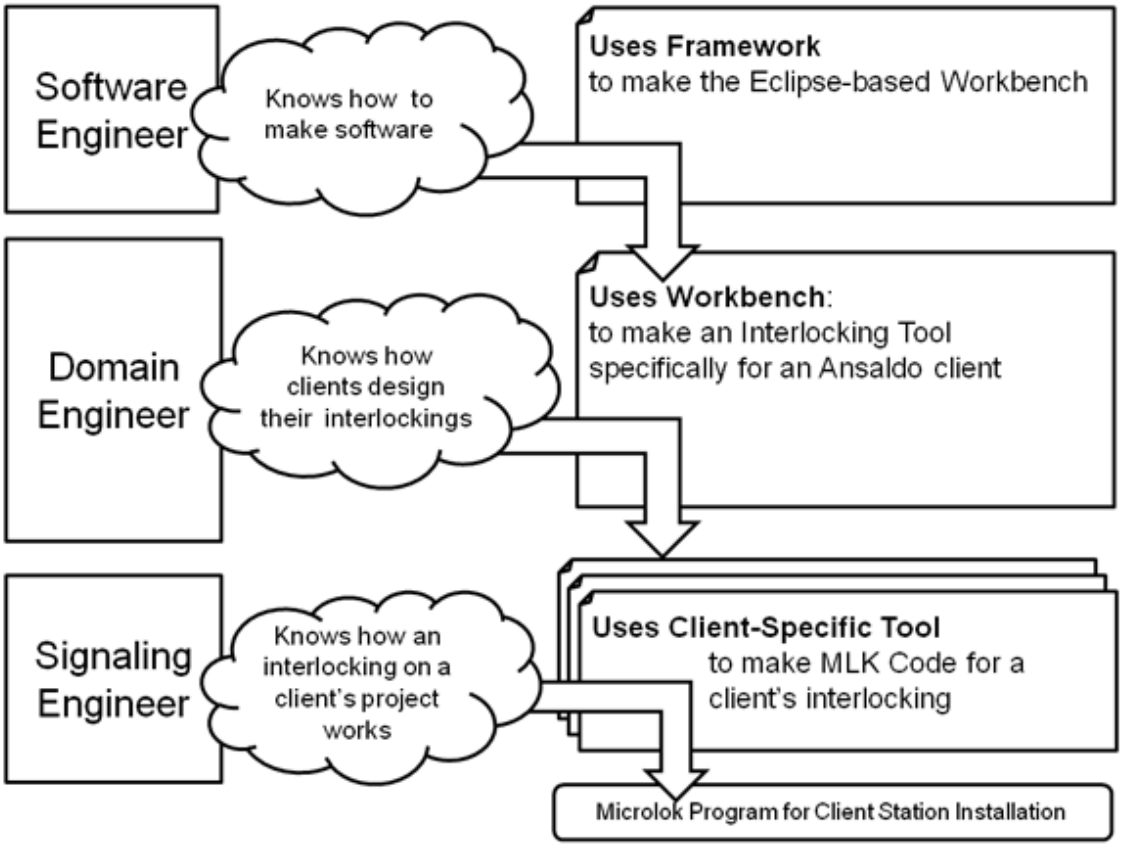
The programming language for Microlok application logic models and mimics a grid of electrical relays. Hence, the names of the program variables follow naming conventions for relays, and the order of the lines of code mimics the arrangement of electrical wires in a relay grid. Signalling engineers use these conventions to read the program and locate information pertinent to specific issues. Therefore, the source code must comply with these conventions for typesetting, ordering, and naming. In Figure 9 these elements are shown under the CodingConvention feature. Conventions have to be defined for each client and project, as there are no common standards. Each company has its own practice in arranging and labelling relays and lines.

Users, Roles and Responsibilities

Up to this point, we have discussed strategies that can manage variability and reuse, but we have not defined who would actually carry out the necessary work. Figure 10 shows the three levels of abstraction of the Mívθα framework, and the corresponding roles and responsibilities.

1. The *software engineer* works on the framework itself, including the features it offers as choices for the definition of clients and the process it enforces on the signalling engineer via the tool. For example, if a new

Figure 10. User roles in the Mint project



type of trackside equipment, for example an "Eurobalise" transceiver, as used in the European Train Control System, is introduced, the feature model has to be extended, so that this type can be chosen in the development of a tool. This leads to a new release of the framework.

2. The *domain engineer* works by making choices in the feature space set out by the software engineer, and by filling in the internals of an empty copy of the framework, to complete it into a product for a specific client of Ansaldo: a client-specific Mint tool. For example for Queensland Rail Regional, the domain engineer would select the features that the regional control table contains, select the notations used in the table, and encode the rules of the QRR ALDS, in terms of the framework's ALDS model.
3. The *signalling engineer* uses the client specific tool provided as the result of the domain engineer's work to generate Microlok application logic from control tables and track layouts. For example, the signalling engineer could enter the Broadlea, Mallawa, and Wotonga stations for the QRR's line duplication project on the high-demand coal line into the QRR-specific Mint tool.

Roles are not rigidly mapped to users. Frequently, the skill-sets of an engineer will allow her or him to work on two levels of the framework. For example, the role of the software engineer needs to be filled with someone who has substantial experi-

ence in the development of model-driven software. This person also models the choices available in defining a tool. From that experience, a software engineer will usually work as a domain engineer when there is no maintenance to be done. A domain engineer will initially be a person trained in the application of modelling tools and programming. However, as the framework matures, programming tasks will become less frequent. Consequently, it will become feasible for signalling engineers to work in this role. Finally, signalling engineers are currently the intended users of the client-specific tools. However, with improved user guidance, it may eventually become feasible to allow clients to prepare their station definitions themselves, while Ansaldo audits them.

Defining a Tool Product

To create a Mint tool for a new client, a domain engineer needs to obtain the ALDS that should include a sample of a fully modelled station complete with track layout, control table, and resulting application logic.

The domain engineer then analyses the control table for the types of sheets and columns it contains. The domain engineer uses an editor that edits an import specification model. Each column in each sheet is imported in two steps. First, the *syntax* of an entry is checked. Syntaxes are defined using a notation, as explained above. A number of notations are built into the framework, but if the client requires new ones, these need to be defined at this time. The syntax check ensures that all entries are lexically and orthographically correct so that they can be meaningfully interpreted. The syntax check is followed by a *semantic* translation to concepts. Semantic translation ensures that the items can be interpreted as a set of requirements for generating code for an interlocking. This phase reveals omissions and inconsistencies. For example, if a locking condition requires a set of points to be normal, but the points are not listed in the points table, this constitutes a semantic error.

In some cases, the conditions for locks can be derived from the track layout using a program. For example, all tracks on the approach side of the signal contribute to the locking condition for an approach lock. In the control table this condition is often defined using a signal-engineering expression. For example '17BT (27AT w 12R)' in the Queensland Rail signalling language defines that the described signal is approach locked if track segment 17B is occupied, or if track segment 27A is occupied while point 12 is set to reverse. The semantic import does not analyse such conditions. It accepts the definitions of the signalling engineers. However, validation of such conditions is possible with the infrastructure provided in Μίνθα.

For each column of the control table, a mapping to concepts has to take place. To achieve this, each column has a small translation program associated with it. These programs are written in the Java programming language. As with notations, ready-made programs can be chosen for a number of column types. A schema model defines which types of translation programs are acceptable to which types of notations.

The domain engineer can now test the import producer by entering the sample station into an Excel worksheet and running the translation to concepts. If an entry is not accepted syntactically, the syntactic conditions may be defined too tightly or the entry needs to be corrected. For the semantic test, the domain engineer has to consult a signalling engineer and validate the concepts that have been produced by hand. The programs that translate the entries produce a log, so inaccurate translation can be traced back to the originating component.

After establishing the concepts, they can be translated to Microlok code. For this, the domain engineer analyses the dependency rules that the ALDS describes. For example, the computation of an approach lock for a signal usually depends on the computation of proved track information, which in turn depends on information from the track circuits. The domain engineer describes the

dependencies and the corresponding code that implements the production of output Microlok code lines. It is common for rules to offer production behaviour that is conditional on inputs. These cases deserve special attention and need to be discussed with a signalling engineer.

To test the process, the domain engineer provides the set of concepts that were computed from the control table, a list of the pieces of track-side equipment that the interlocking should control, and a list of available control inputs like track circuits and train control system route calls. The generation algorithm then seeks the rule that produces the control output. If it finds it, it checks if the existing program already contains a representation of the prerequisites of the rule, or if the prerequisites can be satisfied directly from the inputs. If that is not the case, it looks for a rule that can produce the prerequisite. Either the process is repeated until all requirements are satisfied, or the program runs out of rules it could attempt. The result is a viable but raw program that consists of unsorted lines of code that use arbitrary variable names and does not have the required header and footer comment sections.

To test the generation system, each derivation trace is reviewed with a signalling engineer. In case the system has chosen a different rule than anticipated by the signalling engineer, the rule system is ambiguous and has to be made more specific. If the system fails to derive a solution, the partial traces indicate where rules will have to be added.

For the last step, the domain engineer has to define the rules that determine the names of variables for use in the source code, the rules to be used to sort the equations, and the boilerplate into which they are to be inserted. The Μίνθα framework currently only includes a facility for the naming of variables.

Architecture of the Framework

From a technical point of view, the Μίνθα framework consists of a set of Eclipse plugins, that are grouped into features. Features are packages of related functionality for installation purposes.

All clients require a model of the track layout and the equipment installation. The scope of equipment and layout is finite, so this model and a corresponding parser for a textual notation are part of the framework and included in every client. The track layout information is imported as the first step in the transformation process.

The specifics of a client are stored in a client-description model. This model is defined by the domain engineer and stored as a resource in the tool. It describes the format of the control table Excel file, the rules that transcribe the information into the control table, and the rules of code generation applied. It is used to control the varying aspects of the client workflow, loading varying models and process steps, as described in the following paragraphs. Currently, this model is interpreted at runtime of the client tool, but it could be compiled.

To input the control table from excel, the client description model defines the format of an excel file (schema), while another model stores the content retrieved from the control table file (instance). The model that stores the content does not vary; it describes a list of tables, which express the cells of the sheets in the workbook. Most cells will contain simple data types, but others contain expressions in a signal engineering language. To describe these languages, a library is provided, that contains a parser and storage model for each expression, and the library is included with each client. The model is an EMF model and the parser is a SableCC parser (Gagnon, 1998).

To build these modules quickly, we have developed a tool that translates SableCC grammar definitions to eCore models, and generates a transcription program that imports the text from the data structure of the parser to the structure of EMF. The client description model governs which internal model is allocated for each cell, which implicitly governs which parser is used to analyse the cell content.

The import into the content model of the control table prevents syntactical errors. Contents that cannot interpreted in the way the column requires are rejected. The second step is a transcription from the import model. It invokes a transcription handler on each column. Here, a Java class is used to iterate over the model and transcribes the information to the model. The domain modeller defines these handlers based on a library of default implementations. During the transcription, simple semantic checks are performed. For example, if an expression references a signal or track that is not defined, the input is rejected.

The target of this transcription is the intermediate model of the control table. The model is defined in two pieces: a core model of common concepts is pre-built as part of the framework libraries and a client-specific extension that is specified by the domain engineer to express any requirements that the core does not cover.

After this phase, the semantic code generation is carried out. For each client, a set of rules defines how to produce outputs for controls, like signal settings and points motor controls. The transformation works backwards from the required outputs towards the inputs. Each rule is listed as an entry in the ALDS model, which is referenced from the client-description model. The implementations of the rules are provided by the domain modeller in the form of Java classes. The outputs of the rules are stored in a model of a Microlok II program that is part of the framework.

After this, variables need to be renamed and the lines of code reordered to conform to the client's conventions for source code presentation. This step does not change the semantics of the program and the framework currently does not implement it. However, the client-description model is already designed to express the conventions.

Finally, the Microlok II program model is written out to text form. This is done using the Java Emitter Templates technology that is part of EMF.

Addressing the Domain's Challenges

The section on page 5 introduced a number of challenges. Mívθα addresses these challenges in the following ways:

- The number of turn-around loops for tests is reduced, as the machine-generated code eliminates errors of oversight. The framework allows testing the transformation earlier in the process and with more and more varied inputs.
- The definition of transformation rules for the ALDS requires a slightly higher degree formalisation of the rules. This helps to consider cases that the example-based approach may not have covered. If the resulting code is not altered after the generation, a change in the ALDS rules can be accommodated by regenerating the software for all stations, reducing the cost for rework.
- The framework explicitly models the variability between clients, with a view to a homogenous approach to projects in the business.

Framework Reuse

The Mívθα framework implementation is quite large because it models aspects of signalling at a fundamental level. While it enables the generation of Microlok application logic, this single application purpose is probably not sufficient to justify the cost of construction and upkeep of the framework. This section provides three reuse applications of Mívθα components: to automate the drawing of track layout schematics, to simulate the movement of trains in a station for testing purposes, and to generate code for other controllers than the Microlok II.

Track Layout Diagrams

Track layouts are currently produced using Computer Aided Design (CAD) drawing programs. The

diagrams show the full detail of the installation and the layout is organised around the physical arrangements of the tracks in the station. These diagrams are important in the physical design of the station, but they are not ideal for analysis and design of application logic, because they contain too much information. For example, to analyse the interplay of main signals with points and tracks, the presence of derailers, the number of detection rods per point, variable speed indicators and releases are not necessary. Μίνθα is able to produce track layouts for stations from its track-layout model. The projections are selective, so they only show specific parts of the network or specific features. The layout is performed by the automated graph-drawing package Graphviz (Gansner, 2003). This program does not draw the track pieces to scale. It spaces them evenly so they can be easily analysed. The drawing tool is experimental and it lacks the ability to bind items to a grid. However, these constraints can be introduced as future work.

Train Animator

Interlockings are tested in practice by simulating the movement of trains in a model of the interlocking. Ansaldo uses the Microlok Interlocking Simulator System (MISS) for this purpose. MISS consists of a screen simulation of the interlocking, a physical Microlok card-file that includes the Microlok CPU board and any other periphery boards, and interface software that interprets the information returned from the Microlok. Signalling engineers working as principles testers place trains in the station simulation and methodically observe the response of the Microlok controller. This approach takes substantial time and is expensive to repeat in case a fault is discovered in the application logic.

The animation is designed along the principles of a game. The animator tries to cause a hazardous situation while the application logic tries to prevent it. The animator works with a maximum of two trains that stand, move, or disappear. A train has to enter from the edges of the layout. In the current version, trains do not make shunting moves or split. The animator uses the track-layout model to determine the possible directions of moves, and a simulation state model to store the state of the game. The animator is driven by the ModelJUnit MBT tool (Utting & Legeard, 2006). ModelJUnit accepts a description of the current state – in this case, the occupation of tracks and setting of the signals and points – and a description of operations – stand, move, appear and disappear – and their enabling conditions – for example, a train will not move past a danger signal. ModelJUnit selects a legal move according to a strategy and then waits for the interlocking to respond.

A drawback of the approach is its inability to function without a corresponding MISS installation. This connection is not trivial due to the technical complexities of the protocols that still need to be implemented. Currently we are simulating the responses of the interlocking by hand.

An alternative to the approach above is the use of boolean logic tools such as satisfiability solvers and model checkers.

Satisfiability solvers, SAT solvers for short, identify an assignment of variables of a Boolean formula that will render that formula true. In the rail domain, a SAT solver can be used to ensure that the Boolean logic formulae implemented in the controller program can never produce a hazardous outcome. SAT solvers have been used successfully on specific examples, but not as general strategy (Haxthausen & Peleska, 2007). One of the leading verification companies in the field, Prover Technology AB, has based their service around a SAT solver (pro, 2007).

Model checkers unfold a state space and can show that specific conditions hold. In the rail domain, a model checker can be used to ensure that all state transitions of the station will result in corresponding computations of the interlocking. Our group has previously applied model checkers to this problem with good success. However, the

model checker has to be adapted to the problem and the interpretation of counterexamples can be challenging =(van den Berg et al., 2007)=(van den Berg et al., 2007). In addition, the approach has size limitations.

Both approaches rely on modelling of the interlocking, which requires substantial expertise and is further away from the established testing protocols. Mint aims to support existing approaches and replace them gradually. Further, both approaches gain additional complexity when the time-based aspects of the controller program are included in the proofs. This usually requires the use of a form of temporal logic to express the constraints, and temporal logic constructs are more difficult to prove than regular Boolean logic.

Code Generation for Other Controllers

The language used to express the Microlok application logic is quite similar to that of other popular controllers like Westlock by Westinghouse and Smartlock by Alstom, which are all descendants of the original Solid State Interlocking jointly developed by British Rail, GEC-General Signal, and Westinghouse Signals Ltd in the UK in the 1980s. Since Mívθα is being developed in partnership with Ansaldo, the focus of our work lies with Ansaldo's components. However, the code generation facility can be adapted to generated relay-based code for other interlocking products.

FUTURE RESEARCH DIRECTIONS

The current version of the Mívθα framework addresses the requirements of an automated approach from models to application-logic code. To the signalling engineer that uses a Mint tool, the process appears as one atomic step with no intermediate products. This design leads to several challenges to the approach, which still require solutions.

Back-End Synchronisation

Without access to and information about the intermediate products of the tool chain, signalling engineers have to make changes either to the control table that serves as input or to the application logic that serves as the output. While signalling engineers often complain about a lack of quality in control tables, the usual response to the discovery of an issue is to update the application logic directly. These alterations pose safety threats as they do not necessarily have the safety properties of code generated from the patterns of the ALDS that have been reviewed by several parties with special care. The altered code may or may not perform in an identical or even improved manner over the automatically generated version. As with other software systems, a single alteration can affect the behaviour of the whole system voiding all safety properties. This would not be an issue if it were possible to prove the functional equivalence of two application logic programs. However, this is a very hard problem, due to the temporal and interleaving nature of the logic introduced by the timers in the system. Further, the computation time required for finding the match between two programs, even if a solver is available, grows exponentially with the length of the program.

Regardless of these limitations, signalling engineers will often prefer the application logic to have a specific form that does not follow from the rules of the ALDS. In these cases, the minimal service that Mívθα can render is to manage changes to the output, so they are at least repeatable. We intend to examine an approach that parses the changed output provided by the signalling engineer and subsequently applies the changes as patches to the output on every subsequent generation run. Since the effect of the patch on the behaviour of the application code cannot be understood by looking at the rules of the ALDS, it will be essential to mark such passages in a way that ensures that they are reviewed by a signalling engineer before they are used further in the development process.

Version Management of Models

The Μίνθα framework uses a number of models to define the requirements of a Mint tool: the control table schema, the set of import rules, ALDS dependencies, ALDS rules, and coding conventions. The tool is then used on models of the track layout and control table. In the course of a project, the ALDS is usually updated to include new requirements, but other parts of the definition of the tool may also change. The control tables of stations also change over the course of a project. This leads to a combinatorial version management problem. Which version of the station was compiled with which version of the tool to produce the application code? Intensive use of Mint tools will require a strategy that identifies the models involved in the production of an output. Current research in model-based comparison tools may provide assistance with this issue.

Impact Analysis

Each step of the generation process of the Mint tool transforms the original input information provided by the signalling engineer. In the section above, we have described the limitations of patching the output of the process. Signalling engineers prefer patching to re-generation because they feel that it is easy to verify its impact. To convince signalling engineers to alter the control table input instead of the application logic output, the engineers must be provided with information that simplifies the task of change verification. In other words, the Mint tool must be capable of describing the impact of a change in terms of changes to the output. While it is possible to trace the execution of the various transformation rules, it would be necessary to present the output to the signalling engineer, as described at the of the section on Back-End Synchronisation.

CONCLUSION

Today's signal-engineering processes are still often based on paper documents produced by a variety of unconnected computer-supported tools like CAD programs, word processors, and code compilers. Artefact descriptions are transcribed by hand between the various tools and issues are tracked separately by the tools. Where software support has been developed for signal engineering, the software is bespoke. The framework described in this chapter applies model-based techniques to the problem of signalling application logic development. This kind of framework supports the integration of these currently disparate tools and processes into a common flexible framework that contains and serves all phases of the signal-engineering process. Μίνθα is a demonstration of how this can work.

Beyond the open research issues, Μίνθα needs validation in industrial application. For our partner this implies training of the software-engineering staff in order to manage, support, and expand the framework, and to inform signalling engineers to explain the operation of the tool. To ease adoption in the workplace, Μίνθα itself would benefit from more documentation and should be expanded by process guides known as wizards and cheat sheets that support the users in the process.

From this perspective, it is beneficial for signalling engineers and the organisations they work for to increase their awareness of software-based signal-engineering frameworks and consider how and when they can be adopted into their processes.

REFERENCES

Baker, P., Loh, S., & Weil, F. (2005). Model-driven engineering in a large industrial context — motorola case study. In Briand, L. and Williams, C., (eds) Model Driven Engineering Languages and Systems, volume 3713 of *Lecture Notes in Computer Science*, (pp 476–491). Berlin / Heidelberg: Springer. 10.1007/11557432_36.

Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., & Grose, T. (2003). *Eclipse Modeling Framework*. Addison Wesley Professional.

Eisenecker, U. W., & Czarnecki, K. (2000). *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley.

Gagnon, E. (1998). *SableCC, an object-oriented compiler framework*. PhD thesis, McGill University.

Gansner, E. R. (2003). *Drawing graphs with GraphViz. Technical report*. Murray Hill, NJ, USA: AT&T Bell Laboratories.

Haxthausen, A. E., & Peleska, J. (2007). A domain-oriented, model-based approach for construction and verification of railway control systems. In *Formal Methods and Hybrid Real-Time Systems*, (pp 320–348.)

IEEE. (1987). *IEEE Standard for a Versatile Backplane Bus: VMEbus*. Washington, DC: IEEE Press.

Kastner, C., Thum, T., Saake, G., Feigenspan, J., Leich, T., Wielgorz, F., & Apel, S. (2009). Featureide: A tool framework for feature-oriented software development. In *Proc. IEEE 31st Int. Conf. Software Engineering ICSE 2009*, pages 611–614.

Object Management Group. (2001). *Model Driven Architecture (MDA). Technical report*. Framingham, MA: Object Management Group.

Object Management Group. (2005). *Unified Modeling Language: Superstructure (version 2.0). Technical report*. Framingham, MA: Object Management Group.

Pasetti, A. (2002). *Software Frameworks and Embedded Control Systems, volume 2231 of Lecture Notes in Computer Science*. Springer.

Prover Technology, A. B. (2007). *Prover iLock*. Company Whitepaper.

Utting, M., & Legeard, B. (2006). *Practical Model-Based Testing: A Tools Approach*. Morgan-Kaufmann.

van den Berg, L., Strooper, P., & Johnston, W. (2007). An automated approach for the interpretation of counter-examples. *Electronic Notes in Theoretical Computer Science*, *174*(4). doi:10.1016/j.entcs.2006.12.027

Weiss, D. M. (2008). The product line hall of fame. In *SPLC '08: Proceedings of the 2008 12th International Software Product Line Conference*, (p. 395), Washington, DC, USA: IEEE Computer Society.

ADDITIONAL READING

Allan, J., Arias, E., Brebbia, C. A., Goodman, C. J., Rumsey, A. F., Sciutto, G., & Tomii, N. (2008). *Computers in Railways XI*. WIT Press.

Banci, M., & Fantechi, A. (2005). Instantiating generic charts for railway interlocking systems. In *FMICS '05*. New York: ACM Press. doi:10.1145/1081180.1081197

Berkenkötter, K. (2007). OCL-based validation of a railway domain profile. In *MoDELS'06*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-69489-2_20

Chevillat, C., Carrington, D. A., Strooper, P. A., Süß, J. G., & Wildman, L. (2008). Model-Based Generation of Interlocking Controller Software from Control Tables. In *ECMDA-FA*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-69100-6_24

Endresen, J., Carlson, E., Moen, T., & Alme, K.-J. Haugen, Øy.; Olsen, G. K. & Svendsen, A. (2008), Train Control Language - Teaching Computers Interlocking, In *Computers in Railways XI*, WIT Press.

Gjaldbæk, T. (2002), *Modelling railway interlocking systems*, Master's thesis, Technical University of Denmark, DTU.

Hon, Y. M., & Kollmann, M. (2006). Simulation and Verification of UML-based Railway Interlocking Designs. In *Automatic Verification of Critical Systems*. INRIA.

Magyla, T. (2001). *Evaluation of EBILOCK 950 interlocking system implementation using analytic hierarchy process in 'Transport*. Vilnius: Technika.

Majzik, I., Micskei, Z., & Pintér, G. (2007). Development of Model Based Tools to Support the Design of Railway Control Applications. In *SAFECOMP*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-75101-4_41

Petersen, J. L. (1998). Automatic Verification of Railway Interlocking Systems: A Case Study. In *FMSP-98*. New York: ACM Press. doi:10.1145/298595.298597

Rástocný, K., Janota, A., & Zahradnk, J. (2004). The Use of UML for Development of a Railway Interlocking System. In *SoftSpez Final Report*. Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-540-27863-4_11

Svendsen, A. Olsen, G. K.; Endresen, J.; Moen, T.; Carlson, E.; Alme, K.-J. & Ш‌ystein Haugen (2008), The Future of Train Signaling, in *'MoDELS'07'*, Berlin, Heidelberg: Springer-Verlag.

KEY TERMS AND DEFINITIONS

Application Logic: A set of temporal logic rules that are implemented in an electric or computer-based interlocking to ensure its safe operation.

Automatic Programming: A technique that generates variants of computer programs based on construction rules and requirements input.

Control Table: A type of specification used in signal engineering to formalise the obligations and operational constraints of trackside equipment for an interlocking.

Interlocking: A set of signal machines preventing conflicting movements through an arrangement of tracks such as junctions or crossings.

Microlok II: A computer-based Interlocking controller manufactured by Ansaldo STS.

Signal Engineering: The engineering discipline concerned with the design and construction of railways and railway control systems, and interlockings.

Simulation: An approach to exploring the possible states of a system using a model of that system. Simulation is used for viability analysis and to discover defects.

Track Layout: A type of specification used in signal engineering to formalise the layout of a station or area of control and define the positioning and types of track side equipment at the site.

Chapter 9
Software-Based Self-Test for Reliable Applications in Railway Systems

Alfredo Benso
Politecnico di Torino, Italy

Stefano Di Carlo
Politecnico di Torino, Italy

Alessandro Savino
Politecnico di Torino, Italy

ABSTRACT

The introduction of computers in the control and automation of railway systems led to the massive use of microprocessor-based devices in almost all critical elements of a modern railway infrastructure (e.g., signaling systems, trains control, etc.). Therefore, microprocessor-based systems play a crucial role in the safety, reliability and security of these infrastructures.

The very strict safety standards, which must be guaranteed in a railway system, make the testing of all electronic components a unique and challenging case study. Software-based self-test represents a very attractive test solution to cope with the problem of on-line and off-line testing of microprocessor-based systems. It makes it possible to deeply test hardware components without introducing extra hardware and stressing the system in its operational condition. This chapter overviews the basic principles of software-based self-test techniques, focusing on a set of best practices to be applied in writing, verifying and computing the final test coverage of high-quality test programs for railway systems.

DOI: 10.4018/978-1-4666-1643-1.ch009

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

Safety-critical railway systems are developed according to the highest Safety Integrity Level SIL 4 (IEC, 2011), as imposed by the European Committee for Electrotechnical Standardization (CENELEC) in the standards applicable to the railway industry (CENELEC 50126, 50128, 50129, railway applications standards available at http://www.cenelec.eu). These standards cover the safety management of electrical, electronic, and programmable systems throughout their lives, from concept to decommissioning. They bring safety principles to the management of systems and safety engineering to their development.

In the last few decades, the control and automation of the railway systems have been increasingly engineered around microprocessor-based architectural solutions that, consequently, started playing a crucial role in the safety, reliability, and security of modern railway infrastructures. The design of the testing mechanisms that have to guarantee the correct behavior of a microprocessor-based railway system, their organization and implementation are unique and challenging case studies.

In the UK, the first document addressing safety issues in railway systems is the "Regulation of Railways Act" of 1889. Although it was published a lot earlier than the advent of microprocessors, it introduced a series of requirements on matters such as the implementation of interlocked block signaling and other safety measures motivated by a railway disaster in that year. One of the evolutions of this document is the "Railway Safety Principles and Guidance", produced by the Health and Safety Executive (HSE) for use by organizations wishing to obtain approval for new or altered works, plants and equipment under the Railways and Other Transport Systems Regulations in 1994 (HSE, 2011). Although these manuals have been now replaced, they set the standard for basic guidelines related to safety measures and may help developers in getting a clearer idea of the challenges that need to be addressed when designing testing techniques for railways applications.

The "Guidance on signaling" manual (HSE, 2011) states:

"[...] 5 – INTERLOCKING [...]

(41) Design and construction of mechanical or relay interlocking to inherently 'fail-safe' criteria are required. Programmable electronic interlocking should be designed to composite or reactive fail-safe criteria, using techniques such as redundancy, diversity and self-testing [...]" (pp. 14).

"[...] 8 DEGRADED OPERATION OF SIGNALLING SYSTEMS [...]

(118) Failure of the signaling system should not result in an unsafe situation being created. However, consideration should be given to the actions necessary to allow the passage of trains to continue while the failure condition is rectified. [...]

(119) The signaling system should be able to be reconfigured so that failed equipment can be isolated and, once the nature of the failure is confirmed, the other parts of the system, which are working correctly, can then be used." (pp. 26)

"[...] DEGRADED CONDITIONS [...]

The signaling system should continue to provide for safe passage of trains permitted to run under degraded conditions. The factors for consideration should include: (a) design for 'graceful degradation' so that correctly working parts of the signaling system may continue to be used safely; (b) protection from failure modes creating unsafe situations; [...]" (pp. 34).

From these guidelines, it is feasible to isolate properties that challenge the work of a railway systems test engineer. A safety-critical railway system should be:

- **Fail-Safe:** Safety is the state of being "safe" (from French sauf), the condition of being protected against consequences of failures, damages, errors, accidents, harms or any other event that could be defined as non-desirable;
- **Real-Time:** Most microprocessor-based railway systems have to meet very strict real-time requirements (and tests are not exempt);
- **Designed for Graceful Degradation:** Graceful degradation is the property that enables a microprocessor-based system to continue operating properly (even in degraded conditions) in the event of the failure of some of its components;
- **Serviceable (Also Known As Supportability):** It refers to the ability of technical support personnel to install, configure and monitor computer products, identify exceptions or faults, debug or isolate faults to root cause analysis and provide hardware or software maintenance in pursuit of solving a problem and restoring the product into service;
- **Cost-Effective:** Commercial-Off-The-Shelf (COTS) or, generally, Off-The-Shelf (OTS) components are increasingly used in mission critical applications to overcome high costs and long time-to-market required for building highly dependable custom systems (Kohl, 1999). However, COTS components are not explicitly designed and fabricated for mission critical applications. Their reliability must be guaranteed at higher levels of the design process (e.g., boards/system level). When considering COTS devices, the reader must take into account that the increasing market demand for high computational performance at lower cost and power consumption continually drive semiconductor vendors to develop new microprocessor and peripherals generations. Every new generation incorporates technology innovations from different research domains, such as microelectronics, digital-circuit design, and computer architecture (Psarakis, 2010). These innovations have serious drawbacks on the reliability and testability of these devices, introducing critical challenges for their use in mission-critical applications.

These properties draw a very specific scenario that deeply affects the choice and the design of the testing environment. A fail-safe system at least requires on-line, recurrent test procedures, which are able to guarantee the correct system behavior during its entire mission time. It implies that all test procedures have to run concurrently with the main application the system is designed for, they must be able to report the location of an error as soon as it is detected, and they can never interfere with the real-time requirements of the system. Moreover, real-time requirements drastically limit the time slots in which the test procedures can take full control of the system resources. It should be always possible to interrupt any test routine or, at least, have direct control over the execution time of each test routine. In many situations, this limitation forces the test designers to subdivide the test into functionally independent modules that can run separately without affecting the fault detection that would be otherwise reduced. The fault detection capability also called fault coverage refers to the percentage of some type of faults that can be detected during the test. If the test is interrupted, this final value can considerably decrease. This consideration may affect the choice of the test algorithms, since, in general, they tend to be "atomic", i.e., they cannot be interrupted without decreasing the final test coverage.

Graceful degradation and serviceability require a "degradable" test program. This means that the test program must take into account the serviceability of the system in the way errors are handled. A faulty system with a low serviceability

could have to wait some time to be repaired. If the system is critical for the proper functioning of the railway service, it might be necessary to maintain it running (in degraded safe conditions) even in the presence of errors. This influences the design of the test procedures in two ways:

1. They have to keep a detailed and continuously updated situation of the error status of each component of the system to allow its reconfiguration and the continuity of the service (even if degraded);
2. The components of the system to be tested or excluded from the test have to be dynamically configurable. This allows preserving the testability of the system even after error detection triggered the exclusion of a faulty component.

The use of COTS is another challenge. The low-level architecture of a commercial device is usually unknown to the system's designer. In most of the cases, this precludes the use of structural testing techniques.

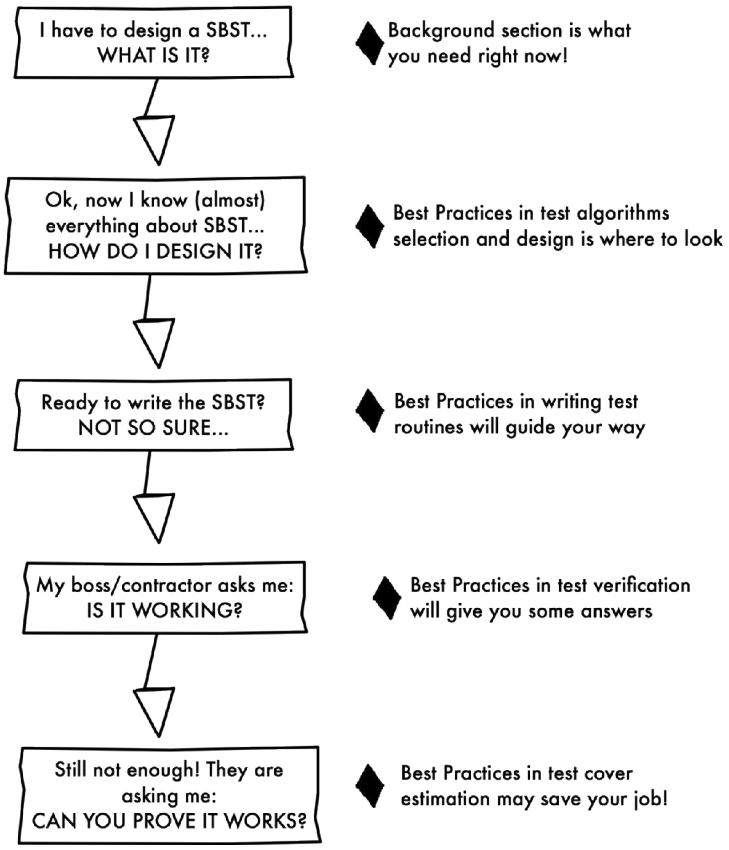
As devices' size decrease, deep-submicron delay defects become more prominent, thereby increasing the need for at-speed tests. Moreover, as multicore processor architectures become more popular, the time needed to test the chip scales with the number of cores, unless the inherent execution parallelism is exploited during testing (Psarakis, 2010). This wide and very specific set of requirements imposes a very flexible but still reliable test approach.

Compared to traditional built-in self-test solutions, Software-Based Self-Test (SBST) introduces several important advantages that nicely fit with the test requirements analyzed earlier (Apostolakis, 2009; Benso, 2006; Psarakis, 2010):

- **It is Non Intrusive:** The test is performed exploiting the Instruction Set Architecture (ISA) of the microprocessor and it does not require the introduction of external hardware that in some applications may be unacceptable in terms of cost or overhead. Minimizing the required hardware also minimizes the power consumption required to perform the test, compared to the consumption of the normal operating mode;
- **It Allows for At-Speed Testing:** Test programs run at the actual operating speed of the processor enabling efficient detection of delay defects that are typical of nanometer technologies;
- **It Allows for In-Field and On-Line Testing:** Self test programs can be periodically scheduled in the field throughout the full life-cycle of the system, thus guaranteeing a continuous monitoring of the health of the system;
- **It Allows for Structural and Functional Testing:** Theoretically any location directly controllable by a microprocessor instruction can be stimulated with any pattern, thus making functional as well as structural test patterns available to the test engineer;
- **It Avoids Over-Test and Over-Stress of the Devices:** Differently from embedded hardware test solutions, SBST avoids overstressing the device under test since it runs in the microprocessor normal operating mode. Only failures that may arise in this working condition are actually stressed. This has potential benefits on the operational life of the devices under test while maintaining elevated levels of reliability;
- **It Allows for on-Site Upgrade of Test Procedure:** Test procedure can be upgraded to meet new standards and reliability requirements, or to adapt to changes or graceful degradation of the system with minimal manual intervention.

SBST in general does not aim at completely substituting traditional hardware test facilities, but rather to supplement them by adding additional test quality at low cost and without the need to perform modifications to the system's hardware. It

Figure 1. Chapter "organization"



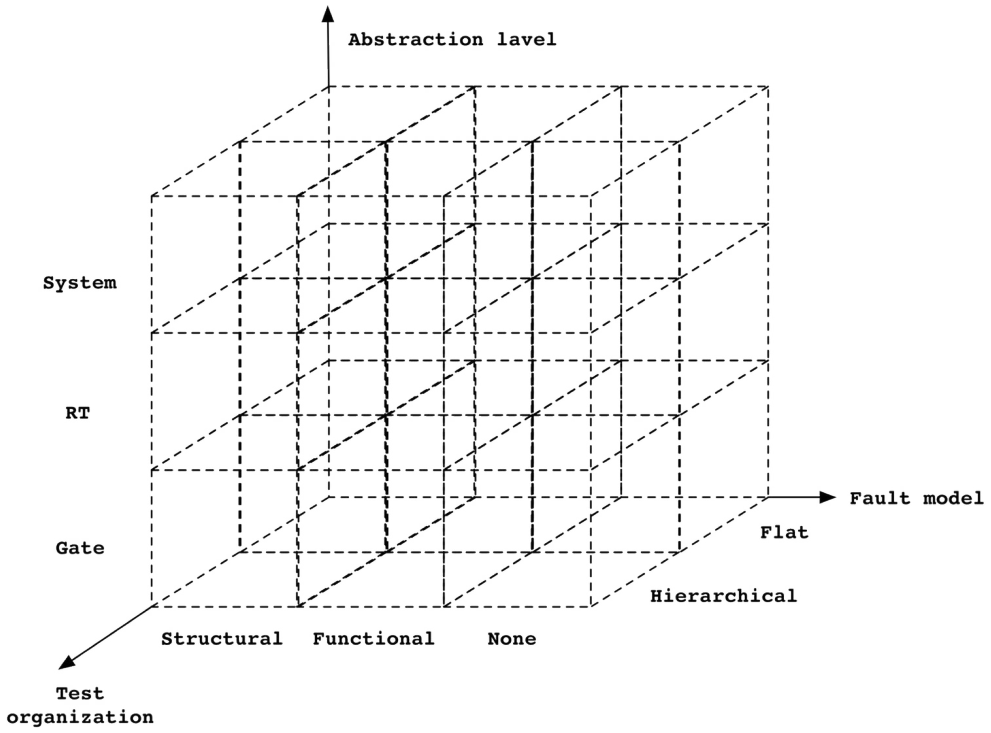
therefore represents a promising solution to allow efficient and reliable use of COTS components while maintaining high dependability standards. This chapter overviews the basic principles of SBST techniques, focusing on a set of best practices to be applied in writing and verifying high-quality test programs for railway systems, and in the definition of their test coverage metrics. We propose them in terms of best practices as a valuable starting point for test-engineers in developing test solutions for their specific applications.

This chapter is intended to be a set of answers (far from a comprehensive theoretical-only approach) to a set of very common questions that often arise during the definition of a test *workflow*. It should be used like a sort of a guide that follows the developer through the SBST development, from the design to the final implementation, validation, and test. Figure 1 gives an idea of what kind of questions we refer to, and where the reader can find the corresponding answers throughout the chapter. A set of additional suggested readings can be found in the conclusions.

BACKGROUND

In the field of microprocessor-based system testing, SBST represents a very attractive solution for testing microprocessor-based systems (Krstic, 2002; Sosnowski, 2006). SBST solutions have the opportunity of enhancing traditional test facilities embedded into COTS components increasing and supplementing them to reach higher test

Figure 2. The SBST design space. Designing a SBST solutions requires to choose the target fault model, the abstraction level at which the system can be described and the overall organization of the final test.



quality. The key idea of SBST is to use on-chip programmable resources to run test programs for the different parts of the system. During SBST, the processor executes self-test programs from the on-chip cache or from system memory, allowing either self-testing or testing of other parts of the system. This essentially eliminates the need for additional test-specific hardware.

Considerable work has been done in the field of SBST of microprocessor-based systems. When defining an SBST method, several design dimensions can be considered, giving the designer a high degree of freedom in the realization of the final test programs. Figure 2 proposes a set of three relevant design dimensions that can be used to classify the most relevant papers on SBST:

1. Fault model.
2. Abstraction level.
3. Test organization.

The fault model dimension defines the type of fault model considered when writing or automatically generating the SBST program. In literature, three main cases have been taken into account. Structural and functional fault models include SBST techniques that aim at generating programs addressing specific types of faults. Structural fault models may include the traditional stuck-at fault model (i.e., node of a circuit stuck at a given logic value), but also higher-level models such as interconnection fault models. Functional fault models are instead models affecting the functionality of the system. Typical examples of these models are those targeting registers and memories deeply analyzed in several publications on memory testing (Di Carlo, 2010). Differently from the first two categories, none refers to a rich set of publications that proposes the generation of test programs without explicitly addressing a specific fault model. The reader must take into

consideration that this dimension only influences the way the test program is constructed without considering the validation and test coverage computation. Even when no fault models are considered during the preparation of the test program, the final test coverage can be evaluated based on specific models. The fault model dimension is strictly related to the abstraction level dimension.

The abstraction level dimension defines the amount of information about the system under test that is available to the test engineer. Typical abstraction levels that have been considered in the SBST literature include the gate level, the Register-Transfer (RT) level and the system or pure functional level.

Finally, the test organization dimension considers how the test is organized with respect to the architecture of the target system. Flat test programs consider the system under test as a monolithic block to be tested, while hierarchical test programs focus on the system's modules one at a time, generating stimuli for each module and then extending those stimuli to the system's level.

Every design choice has its specific advantages and drawbacks. For example, SBST programs based on functional fault models are interesting because they can be designed even when only system level information about the system is available. On the other hand, the lack of this information could limit the capability of attaining comprehensive test coverage. It is worth to mention here that not all combinations of these design choices can be chosen, e.g., structural fault models usually require gate or RT level descriptions and are difficult to apply to system level models. Based on these design dimensions, in this section we will briefly review the most relevant SBST methods proposed in the literature.

During the late 70's and early 80's, an SBST method based on a functional model of the microprocessor and its related functional fault models was proposed by Abraham, Thatte and Brahme (Thatte, 1980, Brahme, 1984). Based on the representation of Figure 2 this solution is classified as [functional, system, hierarchical]. Many microprocessor functional test methods proposed since then where based on this model (Verhallen, 1992; Kranitis, 2003; Paschalis, 2004; Gizopoulos, 2004). Benso (2006) further extended the model proposed by Brahme (1984) considering, together with the different functional blocks composing the microprocessor, a new class of components identifying the interconnections among functional blocks thus adding interconnection test capability to SBST programs.

Contrary to the previous methods, some researchers proposed that, based on real case studies, the only suitable approach for functional testing of microprocessors was an extensive test of the representative functionality. This conclusion produced a set of microprocessor SBST methods that can be classified as [none, system, flat] according to Figure 2. They basically resort to randomly generated sequences of instructions that do not require the use of any fault model. Shen (1998) proposed Vertis, a tool for the automatic generation of programs to be used both for self-test and verification of microprocessors. The test generation is purely functional and does not consider the structure of the microprocessor. The information used to generate the test is the ISA of the microprocessor; it therefore works at the system's abstraction level. The full microprocessor is considered as a single flat component. The generation is based on the concept of code randomizers. Basically, the tool generates pseudorandom instruction sequences for each instruction under test. A pure pseudorandom approach can be enhanced by user-defined sequences, constraints and heuristics. The main drawback of this approach is the size of the generated test programs. Even if the approach is purely functional, the authors show that the structural fault coverage is quite high compared to what can be obtained with a structural test. Pravathala (2002) proposed a similar approach. It presented FRITS (Functional Random Instruction Testing at Speed), an SBST test program generator for microprocessors able to generate random sequences of instruc-

tions using pseudorandom data. One of the main contributions of this paper is the definition of the constraints a test program must fit in order to be fully contained in and executed from the microprocessor cache. This represents a key constraint for the self-test execution in railway systems as will be discussed later in this chapter. Bayraktaroglu (2006) investigated the potential of the application of SBST programs to the test of Sun microprocessors. This work provided strong evidence of the benefit of using SBST in the manufacturing flow of industrial microprocessor designs.

The availability of RT or gate level descriptions of the system allows test engineers to generate test programs increasingly coupled to the architecture of the system. Corno (2004) exploited the use of an evolutionary algorithm for the automatic generation of test programs. The main advantage of this approach, with respect to pure random generation of instruction sequences, is the possibility of introducing a feedback obtained through the computation of a structural fault coverage (e.g., stuck-at fault coverage) required to address corner cases in the test programs. This method can be classified as [structural, RT/gate, hierarchical], in Figure 2. The same overall idea of using structural information to increase the coverage of the generated test programs is exploited in several publications that can be all classified as [structural, RT/gate, hierarchical].

Gurumurthy (2006) described a method for generating sequences of instructions targeting hard-to-detect faults that escape randomly generated programs. Faults are considered at the level of single modules, and for each module an Automatic Test Pattern Generator (ATPG) is used to pre-compute a set of test patterns that are then mapped on the microprocessor ISA. This approach requires the availability of a gate level or RT level description of the system, with a related ATPG. Experiments on an OpenRISC 1200 processor show that the stuck-at fault coverage of a purely random-generated program can be increased from 64% to 82% using this method.

Wen (2006) proposed a hierarchical test generation method that exploits simulations, together with the use of an ATPG, to generate test programs for each module of a microprocessor.

Krstic (2002) proposed a test method that targets both stuck-at faults and path-delay faults. For stuck-at faults they use a hierarchical approach, deriving a set of spatial and temporal constraints for every component (using the microprocessor ISA) used to feed an ATPG or a random generator. The set of test programs is then derived from the generated test patterns. A similar approach is used for path-delay faults.

Although the classification proposed in Figure 2 shows well-separated classes of methods, some publications also proposed hybrid approaches. Kranitis (2005) proposed a component-based divide-and-conquer hierarchical approach that can be classified as [functional/structural, any, hierarchical]. The generation of the test programs is based on the knowledge of the microprocessor's ISA and on the availability of the description of its architecture at any of the abstraction levels (system, RT or gate). The generation process includes four phases: partitioning the system into components, classification of each component, ranking of the components in order to identify the order in which they have to be tested, generation of the test programs. Deterministic test routines are developed based on the specific component (e.g., ALU, cache, etc.). This approach has been further enhanced by Gizopoulos (2008) to test the pipeline logic of sophisticated modern microprocessors.

Chen (2007) proposed a two-step test-program generation that uses multiple-level abstraction descriptions of the microprocessor under test, including the ISA, RT level model, architectural models, and synthesized gate level descriptions. In the first phase, the microprocessor's components are identified; in the second phase test routines are developed for each component exploiting the different abstraction level descriptions. This

approach can be classified as [functional, any, hierarchical].

Some publications addressed the SBST of specific portions of a microprocessor-based system. Di Carlo (2010) and Alpe (2008), for example, addressed the specific problem of testing the cache memory of a microprocessor. The cache represents a key element from the SBST standpoint since it is often used to contain the test programs. The two papers provide a methodology to translate march test algorithms (Di Carlo, 2010) designed to test random access memories into equivalent test programs for cache memories (both the instruction and data cache are considered). Additional publications on SBST of specific microprocessor units can be found in (Raina, 1998; Utamaphethai, 1999; Almukhaizim, 2001; Kranitis, 2005; Fazeli, 2005).

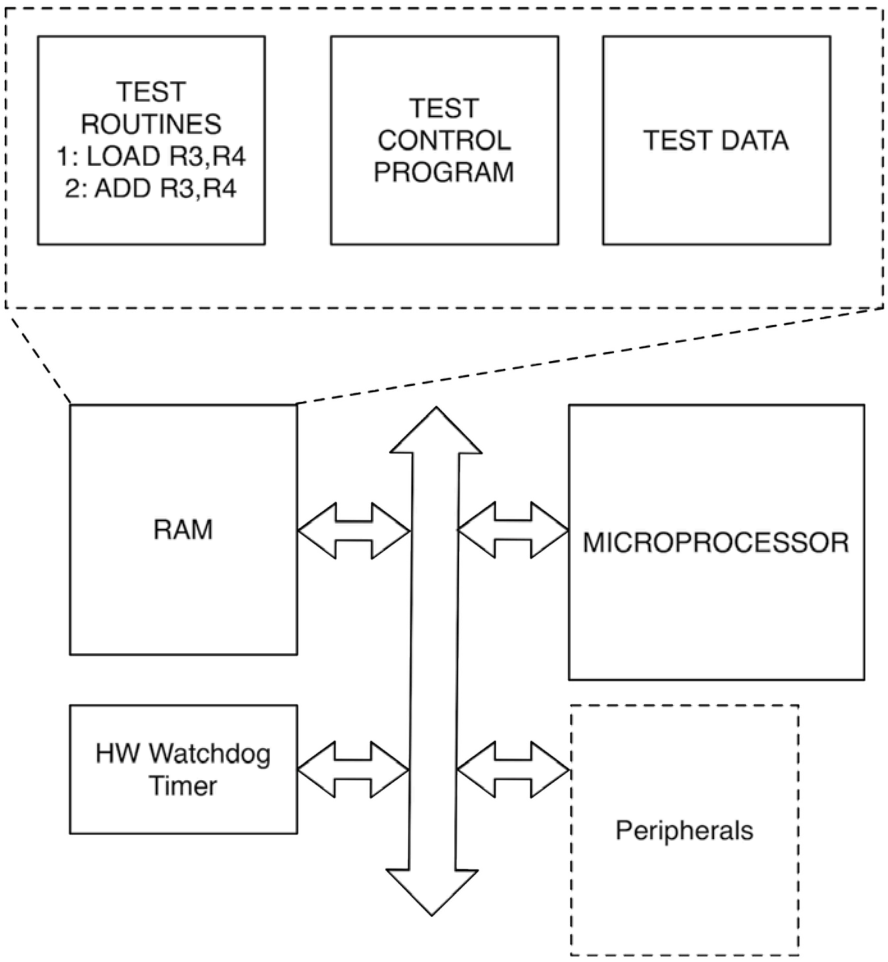
While most of the works on SBST focused on the self-test of microprocessors, very few publications considered the SBST of external peripherals components. Bernardi (2006) proposed an experimental evaluation of the effectiveness of a purely software-based approach, which can be easily and inexpensively implemented on existing SoCs. It presented results on a case study inspired by a real-life application, which exploits a network of SoCs based on the Motorola 6809 processor core. Reported experiments show that the approach achieves relatively high fault coverage with relatively reduced performance penalties.

BEST PRACTICES FOR THE IMPLEMENTATION OF SBST IN RAILWAY SYSTEMS

This section proposes a set of best practices to be used in the design, implementation, and verification of a SBST flow. The suggestions presented in this section are the result of both a deep analysis of the available literature and the authors' own experience in several years of development of SBST procedures for safety-critical microprocessor-based systems with Ansaldo STS, one of the main Italian railway companies. The proposed best practices are not intended for a specific railway infrastructure. They aim at being as practical as possible, but still general enough to cover a wide range of possible applicative scenarios.

The design, implementation and verification flow of an SBST environment can usually be decomposed in several steps. The first step is the system modeling. As explained in the previous section, the design of an efficient SBST solution requires a formal description of the system under test. Since a "flat" approach is not compatible with many of the safety and real-time requirements of railway applications, a modular and hierarchical approach must, in general, be used. Decomposing the whole system in a set of simpler functional units is a very effective way to address the problem (Brahme, 1984). Functional units also include the communication links and connections that allow the system to function (Benso, 2006). The abstraction level at which each unit is described depends on the available information about its internal structure and functions, and also on the desired test program. After each functional unit of the system is identified and modeled, it is necessary to find, for each of them, the best possible test algorithm. This step requires selecting or defining the fault models that the algorithm has to address, and, based on this decision and on the other constraints imposed by the safety and real-time requirements of the system, choosing a proper test algorithm from the ones available in the literature, or developing one ad-hoc. Once a test algorithm has been selected for each functional unit, it must be actually implemented. This operation, often overlooked in research papers, is of paramount importance because it has to avoid a considerable number of technical traps that endanger a successful outcome of the test effort. Finally, verification and coverage evaluation are as important as the previous steps. Verification of the final set of test routines is essential to make sure that the test covers all requirements and is

Figure 3. Typical architecture of a microprocessor-based system required implementing a generic SBST solution



free of programming bugs of any type. Coverage evaluation is also critical, because, without a proper evaluation metric, there is no way for the test engineer to "certify" the efficiency of the test. Coverage evaluation is particularly difficult when functional test is used, since functional test programs are often designed around fault models weakly connected to the actual hardware faults.

Figure 3 shows a very high level overview of how an SBST program is actually applied to the target system, identifying the set of key elements that must always be available in the architecture. All test routines implemented to test the different modules of the system are usually stored in the system's memory, together with the related data. A special program named here "test control program" is the portion of code in charge of scheduling and executing the different test routines. The architecture of Figure 3 perfectly fits the general architecture of a microprocessor-based system with a single hardware constraint that is mandatory in mission-critical applications. Since the test is a software routine any hardware fault that totally prevents the execution of the software, or that forces the software into an infinite loop may represent a catastrophic event for the system. A hardware watchdog timer initialized every time a test routine is executed must be used to detect programs that take an unexpected amount of time to complete their execution. Given the critical importance of this component, whenever possible it

must be designed resorting to fault-tolerant design principles.

Assuming that a proper model of the system has been defined and that, for each functional unit, a proper fault model has been selected, the following subsections suggest a set of best practices that can help test engineers in choosing the best algorithms for each functional unit, implementing and verifying them, and finally in evaluating the final coverage of the resulting test routines.

Best Practices in Test Algorithms Selection and Design

The main limitation in the test algorithms selection is imposed by the fact that, due to the real-time requirements of the system, the time available for the execution of the test algorithms must be partitioned in time-slots, which are determined by the application (Figure 4).

From the SBST point of view this means that the execution of one or a part of the test routine requires three steps:

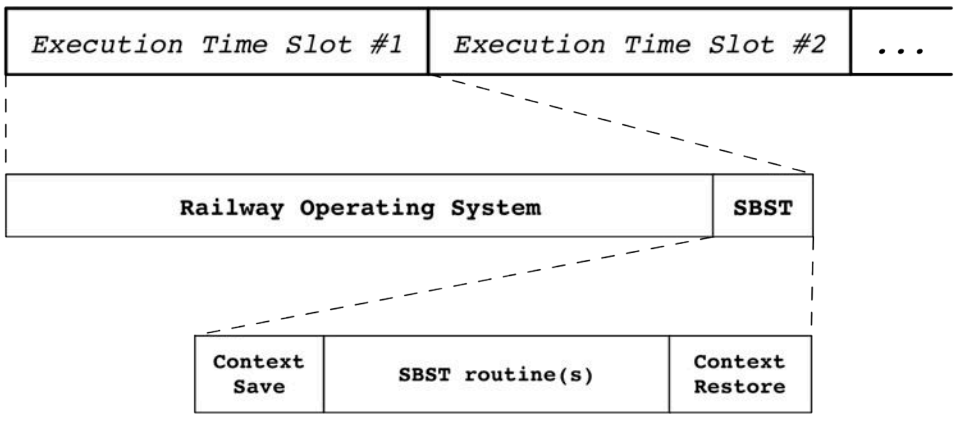
1. The context saving, to save the status of the system.
2. The test procedure execution.
3. The context recovery, to restore the system status and resume normal operations.

Context management cannot be overlooked, otherwise either the test procedure would fail the real-time requirements, or the test would affect the system status, probably making more damage than anything else. The time required to save or recover the system context is related to both the amount of resources that are going to be modified by the SBST procedure and the way they are stored. Since these operations can be time consuming and system-dependent, they may considerably impact the time actually available for the execution of the test routines.

Best Practice 1. Test algorithms must be independent from one another.

As already discussed in detail in the previous sections, most railway critical systems are designed to allow a graceful degradation of their functions in case of error. From the test point of view, this means that not all parts of the system are under test during the mission time. Faulty units that have been excluded from the system must be also excluded from the overall test strategy. This implies that the set and the sequence of the test routines must be dynamically rearranged

Figure 4. Typical time-slot organization in a railway execution environment



depending on the test results (subsystem failure) or on the system's upgrade/downgrade (adding or removing system components). In practical terms, this means that test routines must be independent from one another, allowing to plug-in new test programs without interfering with the existing ones.

Best Practice 2. Atomic test algorithms should be avoided.

Atomic algorithms have to be executed from beginning to end without interruptions. Unfortunately, since two of the most important railway environment constraints require the test to be partitioned in time-slots, atomic algorithms must be avoided since their partitioning may decrease the final fault coverage. Only atomic algorithms whose duration is lower than the minimum time-slot could be theoretically selected. Nevertheless, the duration of the time slot available for testing during the system mission time might change, making this choice a potential hazard.

Best Practice 3. Test algorithms must be partitionable.

The time-slot requirement limits the choice of non-atomic algorithms. In fact, if a test algorithm is made of a sequence of small atomic sequences of instructions, it is important to make sure that the longest atomic sequence is compatible with the worst-case time-slot scenario. As an example, Paschalis (2001) proposed a set of SBST techniques for the ARM9TDMI microprocessor. Since several modules compose the Arithmetic Logic Unit (ALU) of the microprocessor, the authors partitioned the ALU test in a set of algorithms, each of them targeting a different module, e.g., the logic unit, the multiplier, etc. The whole test still targets the ALU but it may better fit a very strict time-slot scenario by scheduling each test algorithm in different time-slots.

Best Practice 4. If there is no other option, also algorithms incompatible with time-slots may be executed.

When there is no other option but to choose an algorithm that is incompatible with the time-slot requirement, the selected test program must be executed at bootstrap or during any non-critical operating time of the system. This means that the test of the corresponding functional unit will not be on-line and that, consequently, that unit will be particularly critical for the reliability of the overall system.

In Di Carlo (2011), we suggest a feasible solution to transform a March test into a SBST of the instruction cache of a microprocessor. Due to the March test structure, the test is not completely atomic and, moreover, the test patterns (composed by instructions) belong to a test that cannot be executed in a single time-slot. Even in this case, the best decision is often to still implement the test and run it as start-up test: it will perform the cache test only every time the system is (re)started, but without strict timing constraints.

Best Practices in Writing Test Routines

Writing the test routines means translating a set of algorithms into executable code. The final code must be optimized to guarantee the best test performance and, in particular, its execution time has to be known and always under control. As for the choice of the test algorithms, timing issues related to the real-time requirements of the system are the main concern in writing the test routines.

Best Practice 5. Test routines must be written in assembly code, it must be encoded to work without external inputs.

Assembly instructions are defined by the microprocessor's ISA. Their execution time and all the resources that are involved in their

execution are known and well documented. High-level languages (such as C, C++, or Java), instead, provide an abstraction layer between the programmer and the system that does not allow a direct control on the instruction execution time at the microprocessor level. Use of high level programming languages may be possible for the test control program (see Figure 3), that is in charge of scheduling the order of the test routines and monitoring the test results, since it usually does not fall under the real-time requirements of the system. Imposing the use of test routines without input allows us to generate test programs that are only influenced by the state of the module under test (fault-free or faulty). Fault-free executions will end-up always in the same predefined sequence of instructions, providing maximum controllability on the test program.

Best Practice 6. Test routines must have full access to the system's resources.

To be able to stress the largest possible set of the system's functionalities, the test routines must be able to execute all instructions of the microprocessor's ISA. Practically, this means that, where different execution modes can be selected (e.g., user and privileged execution mode), the SBST test routines must be granted the one with the highest privileges.

Best Practice 7. Test routines have to run with operating system privileges.

For the same reasons mentioned above, if the system is running an Operating System (OS), the test routines have to run as system processes in order to have the highest control on the system's resources. This implies that the test routines cannot be implemented without a deep knowledge of the OS that will schedule and control their execution.

Best Practice 8. Whenever a sequence of instructions must be executed in different portions of the test program, it must be defined as a macro and not as a function, even if parameters are required.

Defining a function allows programmers to reuse a portion of code wherever it is required. Each function is saved in memory and the test program calls it when its label is found during the program execution. High-level programming languages commonly hide how a function call is actually implemented. It usually requires stack allocation, memory context saving and recovery in order to preserve the microprocessor resources at the function calling time. Therefore, the execution of a function requires interactions with the microprocessor that may lead to modifications of the status of the system under test. Instead, a macro is a sequence of instructions that is physically inserted in the code at compiling time, and therefore repeated every time it is used. No "hidden" behaviors are implied with the execution of a macro. This approach obviously leads to a bigger code, with the advantage of better performances and the full control over the behavior of the code itself and the microprocessor resources.

Best Practice 9. The use of interrupts, system calls, and third-party library functions should be avoided.

It should be clear at this point how the execution time of SBST routines is the critical issue for the test developer and it requires a careful planning and control of the system's status and resources. Any solution that introduces unpredictability over the test execution time or the used resources should be avoided. In particular:

- Interrupts can affect the final coverage by modifying the system state or interrupting atomic operations. Their use should

be limited to the strictly necessary, e.g., as hardware error signaling resources.

- System calls may introduce several undesired side effects. In particular, real-time requirements make many system-calls and almost all OS level services unusable in the test routines, since their scheduling and execution time is neither directly controllable nor deterministic.
- Variables and complex data structures may be a problem because they require access to the external memory, and this operation may introduce uncertainty in the execution time. The use of the external memory to exchange data should be limited to the beginning or at the end of the test routine, to initialize the test or to provide the test results. All other variables used in the test should be allocated in the microprocessor registers or explicitly stored in the code as the instruction operands.

Best Practice 10. Memory virtualization should be avoided

Algorithms for memory testing require a very tight control on the physical addresses used to stress the memory array during the test. Therefore, virtualization techniques have to be avoided because they do not allow full control on the actual physical addresses used during the execution of the code.

Di Carlo (2011) resorts to a March test to implement a SBST of a cache memory, including the tag array in the test. Dealing with the tag part of a cache memory implies the usage of memory addresses that act as test patterns (derived from the original March test). In such cases, the physical translation modifies the patterns that are stored in the tag array, affecting (completely or partially) the test coverage.

Best Practice 11. Whenever possible the test routines (code and data) should entirely fit into the microprocessor cache memories (Pravathala, 2002).

Since cache's dimension is growing very fast, it is now feasible to run a software of ten KBs directly from cache: it only requires the software being stored in contiguous memory addresses to avoid cache misses and consequent line replacements. However, it is not only a matter of time (which still drastically decreases allowing more code to run in the same time-slot): if during the execution the test is in the cache, the microprocessor does not require any external potentially faulty resource (e.g. the RAM). In addition, best practice number 3 suggests that a test algorithm should be modular: if it is so, each module may also be adapted to the cache size dimension limit, allowing each test module to run from the cache itself.

Best Practice 12. Error handling must be quick, useful, and safe.

It is very important to plan how the test environment has to react when an error is detected. Each error may have different priorities and/or meanings but, in any case, it must be handled as quickly as possible. Error handling also means providing the system with a detailed diagnostic report to allow the controlled degradation of its functionalities and, if possible, its later repair. The test control program with a tied collaboration with the OS usually accomplishes error handling.

The amount of diagnostic information must be carefully planned. It may include the status of different sets of internal resources selected depending on the detected error. The important point is to collect all diagnostic data as soon as the error is detected and saving this information, whenever possible, into already tested storage components.

Best Practice 13. Diagnostic data are very important and must always be collected.

A very efficient way of maintaining reliable diagnostic information about the system state is to use the Control Vector technique proposed in (Benso, 2006). A *Control Vector* (CV) is an array of bits that can be accessed by both the test routines and the test control program. Each test routine is associated with one or more bits of the CV. Each routine (or part of it) sets the corresponding bit according to the presence or the absence of the errors it was designed to detect. In general, one control vector can be used as follows:

1. When the entire test starts, the *error-present* value is written in the control vector, e.g., all '1' value.
2. If no error is detected at the end of a test routine, the corresponding bit is set to the *error-free* value ('0').

This mechanism assures that all routines are executed, and reports an error condition if a routine, for any reason, is not executed. It allows the test control program to exactly know where the error is located (in terms of routines or portions of routines). The control vector can be software or hardware implemented. In the latter case it can behave as an interrupt register, to immediately trigger the execution of the handling routine in case an error is detected.

The control vector (CV) can even be implemented as a hierarchy of arrays, where each CV is linked to a set of control sub-vectors, each of them controlled by a subset of test routines. Each sub-vector contributes by setting a single bit of its parent vector (i.e., by OR-ing its content bits), allowing a recursive, tree-based error detection. The main test program is able to quickly check the test status by simply reading the top level CV. Further diagnostic investigations of the CV hierarchy may be then handled by the test controller during the error handling procedures. This solution has an additional advantage because the hierarchical structure well fits with the modularity of the physical system, the active components of which can be dynamically reconfigured following upgrades or graceful degradations.

Figure 5 shows the implementation of a two-level hierarchical CV. Control vectors can be allocated in several locations: registers, memory locations, stack locations and so on.

The control vector alone cannot catch all the information required to analyze errors. The system context must also be available to understand the error sources.

The *Error Context Data* (ECD) can be defined as the set of values of the internal resources that are believed to be key-points in understanding the causes of the error. When writing the code to collect this information, the test engineers must consider the following points:

1. The context data has to be collected at the error detection time or, at least, as soon as possible.

Figure 5. Control vector implemented as a hierarchy of arrays

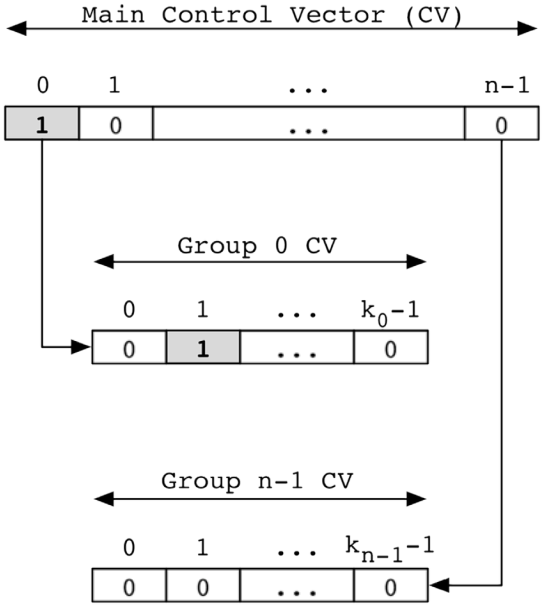
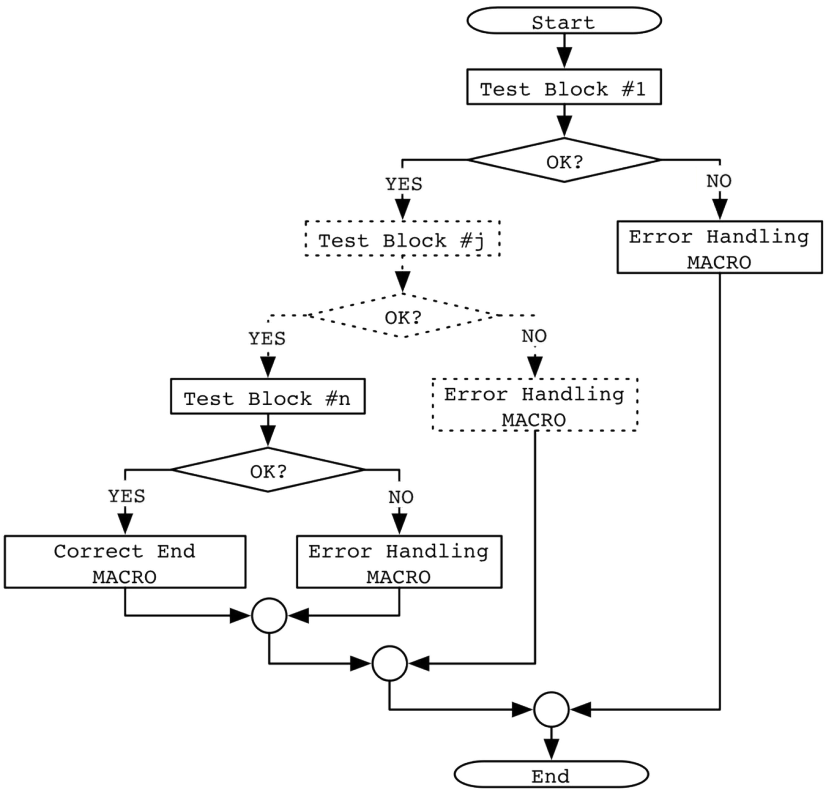


Figure 6. Generic test algorithm execution flow



2. The context data saving should not alter the system's state that caused the error and should be atomic.
3. The context data may be classified as: (a) common data, i.e., system configuration register values, stack pointers, program counter; (b) routine-specific data, i.e., value mapping of all used registers. Both classes must be identified and stored. Any lost information will affect the error analysis, reducing the possibility of diagnosing the error source(s).
4. The stored context data should be safe. Data redundancy techniques should be used if necessary, e.g., signature, error detecting/correcting codes, CRC, etc.

Best Practices in Test Verification

Once the complete test program is implemented, it is necessary to verify that the final implementation perfectly corresponds to the original test algorithms. The verification of SBST routines can be particularly difficult, especially when dealing with the real-time constraints. Nevertheless, the reader must consider that, following the best practices discussed so far allows us to generate SBST programs with some key peculiarities that make the verification task easier than for other types of software. First of all, the code is written in assembly language that provides a direct control over timing and system resources. Moreover, it has no inputs, i.e., its execution flow can be predicted and is not affected by a workload. It is organized as a sequence of

basic and independent blocks of instructions with the goal to only excite different error conditions and, in the presence of an error, to report the event as soon as possible.

Figure 6 shows the basic structure a test program should look like. There are only two exit conditions: one with no errors, and one whenever an error is detected. The error handling code is placed at the bottom of the test program code and is the same for every block. Moreover, the SBST code has no complex data structures (best practice 9), it runs in a very constrained environment, and endless loops are always avoided by the presence of a hardware watchdog timer, as previously discussed.

In general terms, three tools are available for software verification: debugging, visual inspection, and code execution tracking. Unfortunately, not all of them can be used when verifying SBST programs.

Best Practice 14. Debugging tools should be avoided.

The main reason for this best practice is that the execution of the test under the control of a debugger is very likely going to be considerably different from a normal execution of the same code. This is due to the fact the most debugging tools are designed on top of particular OS services. During debug, these services may have some limitations in accessing some of the resources under test, making the debug and verification of the test routines impossible or not reliable. Moreover, modern microprocessors implement instructions parallelism at run-time, and the debug tools are usually not able to handle the interdependence between instructions. For these and other similar reasons, unless the user has a full and deep knowledge of how the debugging tool works, its use should be avoided. This best practice introduces several problems in the validation of tests that must be solved using different strategies.

Best Practice 15. Several people must perform visual inspection.

Visual inspection of the code means to manually and visually check the correspondence between the original algorithms and the implemented assembly code. To maximize error detection, more than one person should perform this operation independently.

Best Practice 16. Fault-free executions are necessary to correct bugs and to evaluate the code performances.

If the test code is executed in a fault-free system, it should not detect any error. If this is not the case, then the error is obviously caused by a bug in the code. This operation usually allows:

1. The detection of the majority of the errors in the code (whenever these errors are not masked).
2. The tuning of the SBST code to better perform in the final operative environment.

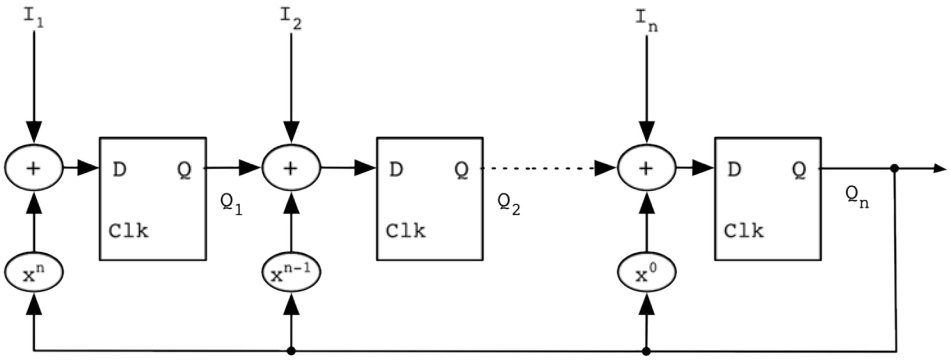
For each test routine, some parameters should be collected for off-line evaluation:

- The context saving time (when the procedure starts).
- The test routine execution time.
- The context recovery time (when the procedure ends).

With this information it is possible to verify the compatibility of each test routine with the time-slot's requirements.

Best Practice 17. A control-flow mechanism has to guarantee that every part of the test program is executed.

Figure 7. MISR architecture



One of the common concerns when using SBST techniques is how to guarantee that every part of the test program is actually executed. This is only possible by implementing a run-time execution flow checking mechanism. This mechanism must not modify the original algorithm and should introduce the minimum performance overhead.

A very efficient way to implement this mechanism is to generate a step-by-step signature of the executed code. At the end of the execution, the verification of the signature is a reasonable guarantee that all required blocks of code have been executed in the correct order. A single signature is enough to identify the test program execution of a fault-free system.

A very efficient and reliable way to generate a signature of this kind is to implement a Multiple Input Signature Register (MISR). In general, a MISR is a hardware component able to compute a signature of the sequence of its input data. An n-bit MISR, as shown in Figure 7, is an n-bit shift register with an n-bit parallel input, where the input of each stage is the XOR between the output of the previous stage and one bit from the parallel input. The value of the input bit of each stage is generated using a primitive polynomial that selects the bits that have to be XOR-ed to produce the new bit. A seed (the shift register initial sequence), and a primitive polynomial therefore define a MISR. The primitive polynomial specifies the mathematical function that generates the signature. Any variation in the sequence of the input values generates a different signature.

MISRs are often used in hardware testing to compute the signature of the outputs generated by a component under test. In an SBST environment a MISR can be easily implemented in software to compute a signature of the executed blocks of code. To do that, each block of code needs to be assigned a pair of unique IDs (one for the beginning and one for the end of the block) that will be used as the MISR input during the signature generation. The MISR is then activated at the beginning and at the end of each block of code (providing the corresponding ID as input). At the end of the test, the content of the MISR can be used as the unique signature of the followed execution flow.

The algorithm reported in Figure 8 shows the amount of operations required to implement a MISR with primitive polynomial $x^{16}+ x^{14}+ x^{13}+ x^{11}+1$ using C language directives. The check of the correctness of the signature can be scheduled at the end of the full test program or even at predetermined locations inside the code. Obviously, if the system (and the test) is reconfigured following a graceful degradation, also the final signature must be updated.

The introduction of the MISR does not affect in any way the coverage of the test routines. Instead

Figure 8. MISR implementation – C code

```
short input;
short misr = 0;
/* Primitive polynomial: x^16 + x^14 + x^13 + x^11 + 1 */
short poly = 0xB401;
.......
input = 0xAF21 /* example of current block ID */
misr = (misr >> 1) ^ input ^ ((misr << 15) ? poly : 0));
```

it guarantees that they have all been executed in the desired sequence.

Best Practices in Test Coverage Estimation

Once the implementation of the test is completed, it is necessary to define some metrics to somehow evaluate the performances of the test program in terms of fault coverage. Without this step, there is no way to "certify" the correctness and validity of the test. The estimation of the test fault coverage requires two steps:

1. The definition of the target fault model.
2. The evaluation of the coverage based on the selected fault model.

Both steps are strictly dependent on the available system description, and on the possibility of simulating the behavior of the system in presence of the target faults. The fault model used to estimate the test coverage does not require being the same used during the test implementation phase.

Best Practice 18. If a fault simulator is available, the test fault coverage based on the fault models it is able to manage must always be computed.

Regardless the fault model adopted to design the selected test algorithm, and even if the theoretical test coverage can be computed, fault simulation represents the best method to assess the final coverage of a test program. In fact, the process of translating an algorithm into its equivalent assembly code (with the considerable set of constraints discussed before) may have affected its coverage capabilities. It is therefore good practice to verify the actual coverage provided by the implemented algorithm.

Fault simulation tools require in general low-level descriptions of the system (e.g., gate level or RT level), and can therefore be employed only when these models are available to the test engineer. When both gate level and RT level descriptions are available, a gate level fault simulation considering gate level fault models (e.g. stuck-at fault, delay faults) is preferable in order to obtain very precise estimations.

Best Practice 19. Whenever only functional descriptions are available and functional fault models are considered, new coverage metrics must be defined.

When only a functional description of the system is available, the test coverage can only be estimated on the base of a set of functional fault models. In this case new functional coverage metrics that exploit the limited set of information about the target system must be defined. A functional coverage metric is a measure of the capability of an algorithm to verify a certain set of functionalities. Several metrics can be considered

(e.g., percentage of ISA coverage, percentage of user resources stimulated by the test, etc.), and the choice of the metric usually depends on the target application as well as on the policy that can be accepted in the environment where the system is going to work.

Following this scenario, Benso (2008) proposed a methodology to define an entity-relationship (ER) diagram to model the behavior of a microprocessor, its instruction set architecture, and a target test program. The overall idea is to realize a software framework able to collect run-time test related information and to store it into a relational database. Collected information can then be used to compute coverage metrics implemented with a set of SQL queries on the data. The level of details stored in the database depends on the type of description of the system. This technique can be also used to improve a test program by identifying poorly covered functionalities that need to be specifically addressed. The microprocessor model can be easily extended to apply the same techniques to more complex microprocessor-based systems.

Best Practices in… Practice

All previous best practices are intended to be a set of practical guidelines answering very common problems in the design of an SBST framework. Often, literature provides good ideas but lacks of implementation and practical details. This chapter is intended to bridge the gap between theory and implementation.

All suggested Best Practices come from in-field r&d activities over the years dealing with *real* test cases requiring *real* working solutions. Although they seem not to be supported by sufficient theoretical proofs, they frequently saved us time and money. Time is saved avoiding the introduction of common issues inside SBST solutions that will be discovered during the final test, usually at the end of the work when time is running out. Money is preserved by both reducing the time-to-market, and by identifying where critical conditions may arise and, consequently, better distribute human resources.

CONCLUSION

This chapter presented an overview of the application of SBST techniques to mission-critical railway systems. The literature on SBST is very full of test approaches, with the first publications dated back in the late ‘70s, and new test methods are continuously presented at international conferences and published in international journals in the field. Proposing new test methods or simply reviewing the existing literature would have made this chapter outdated in a short time frame. A set of best practices was presented for test engineers to follow in a real industrial setup.

Most of the best practices presented throughout this chapter are the results of the experience of the authors in several years of collaboration with Ansaldo STS in the development of SBST programs for COTS microprocessors. They represent a valuable instrument to fill the gap between the academic research community, which is continuously proposing new test algorithms, and the industry that needs to adapt these solutions to its specific industrial setup.

REFERENCES

Almukhaizim, S., Petrov, P., & Orailoglu, A. (2001). Low-Cost, Software-Based Self-Test Methodologies for Performance Faults in Processor Control Subsystems. In *IEEE Conference on Custom Integrated Circuits* (pp. 263–266). San Diego, CA, USA: IEEE Computer Society Publications.

Alpe, S., Di Carlo, S., Prinetto, P., & Savino, A. (2008) Applying march tests to k-way set-associative cache memories. In *13th IEEE European Test Symposium* (pp. 77–83). Verbania, Italy: IEEE Computer Society Publications.

Apostolakis, A., Gizopoulos, D., Psarakis, M., & Paschalis, A. (2009). Software-based self-testing of symmetric shared-memory multi-processors. *IEEE Transactions on Computers*, *58*(12), 1682–1694. doi:10.1109/TC.2009.118

Bayraktaroglu, I., Hunt, J., & Watkins, D. (2006). Cache resident functional microprocessor testing: Avoiding high speed IO issues. In IEEE *International Test Conference* (pp. 1-7), Austin, TX, USA: IEEE Computer Society Publications.

Benso, A., Bosio, A., Prinetto, P., & Savino, A. (2006). An on-line software-based self-test framework for microprocessor cores. In *International Conference on Design and Test of Integrated Systems in Nanoscale Technology* (pp. 394–399). Tunis, Tunis: IEEE Computer Society Publications.

Benso, A., Di Carlo, S., Prinetto, P., Savino, A., & Scionti, A. (2008). Using ER models for microprocessor functional test coverage evaluation. In *11th International Biennial Baltic Electronics Conference* (pp. 139–142). Tallin, Estonia: IEEE Computer Society Publications.

Bernardi, P., Bolzani, L., Manzone, A., Osella, M., Violante, M., & Sonza Reorda, M. (2006). Software-based on-line test of communication peripherals in processor-based systems for automotive applications. In *Seventh International Workshop on Microprocessor Test and Verification.* (pp. 3–8). Austin, TX, USA: IEEE Computer Society Publications.

Brahme, D., & Abraham, J. A. (1984). Functional testing of microprocessors. *IEEE Transactions on Computers*, C-*33*(6), 475–485. doi:10.1109/TC.1984.1676471

Chen, C.-H., Wei, C.-K., Lu, T.-H., & Gao, H.-W. (2007). Software- based self-testing with multiple-level abstractions for soft processor cores. *IEEE Transactions on Very Large Scale Integration (VLSI). Systems*, *15*(5), 505–517.

Corno, F., Sanchez, E., Sonza Reorda, M., & Squillero, G. (2004). Automatic test program generation: a case study. *IEEE Design & Test of Computers*, *21*(2), 102–109. doi:10.1109/MDT.2004.1277902

Di Carlo, S., & Prinetto, P. (2010). Models in Memory Testing, From functional testing to defect-based testing. In Wunderlich, H.-J. (Ed.), *Models in Hardware Testing* (pp. 157–185). Springer DEU.

Di Carlo, S., Prinetto, P., & Savino, A. (2011). Software-based self-test of set-associative cache memories. *IEEE Transactions on Computers*, *60*(7), 1030–1044. doi:10.1109/TC.2010.166

Fazeli, M., Farivar, R., & Miremadi, S. (2005). A software-based concurrent error detection technique for PowerPC processor-based embedded systems. In *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (pp. 266–274). Monterey, CA, USA: IEEE Computer Society Publications.

Gizopoulos, D., Paschalis, A., & Zorian, Y. (2004). *Embedded Processor-Based Self-Test*. Springer Press.

Gizopoulos, D., Psarakis, M., Hatzimihail, M., Maniatakos, M., Paschalis, A., Raghunathan, A., & Ravi, S. (2008). Systematic software-based self-test for pipelined processors. *IEEE Transactions on Very Large Scale Integration (VLSI). Systems*, *16*(11), 1441–1453.

Gurumurthy, S., Vasudevan, S., & Abraham, J. A. (2006). Automatic generation of instruction sequences targeting hard-to-detect structural faults in a processor. In *IEEE International Test Conference* (pp. 1–9). Austin, TX, USA: IEEE Computer Society Publications.

International Electrotechnical Commission. IEC (2011). Functional Safety and IEC 61508. Retrieved March 25, 2011, http://www.iec.ch/functionalsafety/

Kohl, R. J. (1999). Establishing guidelines for suitability of cots for a mission critical application. In *Annual International Computer Software and Applications Conference* (pp. 98-99). Phoenix, AZ, USA: IEEE Computer Society Publications.

Kranitis, N., Paschalis, A., Gizopoulos, D., & Xenoulis, G. (2005). Software-based self-testing of embedded processors. *IEEE Transactions on Computers*, *54*(4), 461–475. doi:10.1109/TC.2005.68

Kranitis, N., Xenoulis, G., Paschalis, A., Gizopoulos, D., & Zorian, Y. (2003). Application and analysis of rt-level software-based self-testing for embedded processor cores. In *International Test Conference* (pp. 431–440). Charlotte, NC, USA: IEEE Computer Society Publications.

Krstic, A., Lai, W.-C., & Cheng, K.-T. & Chen &. & Dey S. (2002). Embedded software-based self-test for programmable core-based designs. *IEEE Design & Test of Computers*, *19*(4), 18–27. doi:10.1109/MDT.2002.1018130

Office of Rail Regulation. HSE (2011). *Railway Safety Principles and Guidance, Part 2, Section D, Guidance on Signaling*. Retrieved March 25, 2011, http://www.rail-reg.gov.uk/upload/pdf/rspg-2d-signlng.pdf

Parvathala, P., Maneparambil, K., & Lindsay, W. (2002). FRITS - a microprocessor functional BIST method. In IEEE *International Test Conference* (pp. 590–598). Baltimore, MD, USA: IEEE Computer Society Publications.

Paschalis, A., & Gizopoulos, D. (2004). Effective software-based self-test strategies for on-line periodic testing of embedded processors. In *Design, Automation and Test in Europe Conference and Exhibition* (pp. 578–583). Paris, France: IEEE Computer Society Publications.

Paschalis, A., Gizopoulos, D., Kranitis, N., Psarakis, M., & Zorian, Y. (2001). Deterministic software-based self-testing of embedded processor cores. In *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings* (pp. 92-96)

Psarakis, M., Gizopoulos, D., Sanchez, E., & Reorda, M. S. (2010). Microprocessor Software-Based Self-Testing. *IEEE Design & Test of Computers*, *27*(3), 4–19. doi:10.1109/MDT.2010.5

Raina, R., & Molyneaux, R. (1998). Random self-test method applications on PowerPC microprocessor caches microprocessor caches. In *8th Great Lakes Symposium on VLSI* (pp. 222–229). Lafayette, LA, USA: IEEE Computer Society Publications.

Shen, J., & Abraham, J. A. (1998). Native mode functional test generation for processors with applications to self-test and design validation. In *International Test Conference* (pp. 990–999). Washington, DC, USA: IEEE Computer Society Publications.

Sosnowski, J. (2006). Software-based self-testing of microprocessors. *Journal of Systems Architecture*, *52*, 257–271. doi:10.1016/j.sysarc.2005.05.004

Thatte, S. M., & Abraham, J. A. (1980). Test generation for microprocessors. *IEEE Transactions on Computers*, *29*(6), 429–441. doi:10.1109/TC.1980.1675602

Utamaphethai, N., Blanton, R., & Shen, J. (1999). Superscalar processor validation at the microarchitecture level. In *20th International Conference On VLSI Design* (pp. 300–305). Goa, India: IEEE Computer Society Publications.

Verhallen, T., & van de Goor, A. (1992). Functional testing of modern microprocessors. In, *3rd European Conference on Design Automation* (pp. 350–354). Brussels, Belgium: IEEE Computer Society Publications.

Wen, C.H.-P. & Wang Li-C & Cheng K.-T. (2006). Simulation-based functional test generation for embedded processors. *IEEE Transactions on Computers*, *55*(11), 1335–1343. doi:10.1109/TC.2006.186

ADDITIONAL READING

Gizopoulos, D., Paschalis, A., & Zorian, Y. (2004). *Embedded Processor-Based Self-Test*. Springer Press.

Psarakis, M., Gizopoulos, D., Sanchez, E., & Reorda, M. S. (2010). Microprocessor Software-Based Self-Testing. *IEEE Design & Test of Computers*, *27*(3), 4–19. doi:10.1109/MDT.2010.5

Thatte, S. M., & Abraham, J. A. (1980). Test generation for microprocessors. *IEEE Transactions on Computers*, *29*(6), 429–441. doi:10.1109/TC.1980.1675602

Zhou, J. (2009). *Software-Based Self-Test under Memory, Time and Power Constraints.* Doctoral dissertation, Institut fur Technische Informatik der Universität Stuttgart, Stuttgart, Germany. Retrieved March 25, 2011, http://elib.uni-stuttgart.de/opus/volltexte/2010/4836/.

KEY TERMS AND DEFINITIONS

Fault Coverage Computation: Fault coverage refers to the percentage of some type of fault that can be detected during the test of any engineered system.

Fault Models: A model of the behavior of defective circuitry in an integrated circuit.

Functional Testing: Functional testing is a type of black box testing that bases its test cases on the specifications of the systems' components under test.

Microprocessor: An integrated circuit that contains the entire central processing unit of a computer on a single chip.

Real-Time Systems: A real-time system is a system that responds in a (timely) predictable way to unpredictable external stimuli arrivals

Safety Critical Application: A system in which any failure or design error has the potential to lead to loss of life.

Software-Based Self-Test: The ability of a system to test its functionalities without the intervention of an external entity, exploiting software routines.

Software Verification: Software verification is a broader and more complex discipline of software engineering whose goal is to assure that software fully satisfies all the expected requirements.

Chapter 10
Real-Time Hardware-in-the-Loop in Railway:
Simulations for Testing Control Software of Electromechanical Train Components

Silvio Baccari
University of Sannio, Italy

Giulio Cammeo
AnsaldoBreda, Italy

Christian Dufour
Opal-RT Technologies, Canada

Luigi Iannelli
University of Sannio, Italy

Vincenzo Mungiguerra
AnsaldoBreda, Italy

Mario Porzio
AnsaldoBreda, Italy

Gabriella Reale
University of Sannio, Italy

Francesco Vasca
University of Sannio, Italy

ABSTRACT

The increasing complexity of modern ground vehicles is making crucial the role of control for improving energetic efficiency, comfort and performance. At the same time, the control software must be frequently updated in order to let the vehicle respond safely and efficiently within more sophisticated environments and to optimize the operations when new vehicle components are integrated. In this framework real-time hardware-in-the-loop simulations represent a fundamental tool for supporting the verification and validation processes of the control software and hardware. In this chapter a railway case study will be presented. The mathematical models of the most relevant electromechanical components of the vehicle powertrain are presented: the pantograph connected to an ideal overhead line with continuous voltage; the electrical components of a pre-charge circuit, the line filter and the braking chopper; the three-phase voltage source inverter and the induction motor; and, finally, the mechanical transmission system, including its interactions with the rail. Then the issues related to the real-time simulation of the locomotive components models are discussed, concentrating on challenges related to the stiff nature of the dynamic equations and on their numerical integration by combining field programmable gate array

DOI: 10.4018/978-1-4666-1643-1.ch010

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

(FPGA) and central processing unit (CPU) boards. The usefulness of the real-time hardware-in-the-loop simulations for the analysis of railway control software will be demonstrated by considering the powertrains of two real metropolitan trains under complex scenarios, i.e., stator winding disconnection of the induction motor, pantograph missing contact, wheel-rail slipping phenomenon.

INTRODUCTION

Hardware-in-the-loop (HIL) techniques are widely used in several engineering fields for testing, verification and validation of specific components designed to perform dedicated functions into complex environments. In our framework, the electronic control unit (ECU), as regards its functionalities dedicated to the control of electromechanical components of a typical train traction architecture, represents the device under test. The ECU is composed of a hardware device and its firmware and it is inserted into the loop completed with mathematical models emulating other parts of the equipment under control. In particular the models are implemented through software programs and corresponding processing units able to provide in real-time the electrical stimuli needed to partially or fully test and exercise the ECU. In this way the ECU and the control algorithms coded in it respond to the simulated signals so as they were operating in the real system under control.

The benefits of real-time HIL systems are manifold. HIL systems allow the testing of new ECUs so as they support the verification, validation and regression testing processes of the control software. Replacing (part of) the real equipment under control with computers running software simulations greatly reduces the size and complexity of applications and increases the flexibility and rate of running of different tests and scenarios. Obviously, this also means faster procedures for the calibration and the maintenance of the control software. Moreover HIL tests can be done without damaging equipment or endangering lives, then determining an improvement in term of costs, duration, feasibility and security.

Since several decades real-time HIL simulations have been applied to transportation systems, traditionally in the aerospace and automotive fields. More recently this technique has shown its potentialities also for railway systems (Terwiesch, et al., 1999; Dufour, et al., 2008; Allegre, et al., 2010) and in perspective it seems useful for supporting the analysis of many other railway control problems (Goodall, 2011). In fact, different railway on board subsystems can be effectively tested and validated through a HIL platform that emulates the behavior of the vehicle. For instance, odometry, anti-skid and braking functionalities could be tested through HIL (Pugi, et al., 2005a; Pugi, et al., 2005b). The same approach can be used also for investigating complex phenomena like adhesion between the rail and the wheel (Malvezzi, et al., 2008) or the interaction between the pantograph and the catenary (Collina, et al., 2004; Facchinetti and Mauri, 2009). The HIL technology has been also successfully applied for testing critical subsystems dealing with the train protection and the communication with interlocking systems (Di Tommaso, et al., 2005). In those cases the platform simulates sequences of real events aimed to test the correct behavior of safety critical railway control systems. With respect to traction subsystems, such applications need a different kind of models, particularly focused on describing temporal relations (concurrency, deadlock, etc.) rather than physical behaviors.

In this chapter we concentrate on real-time HIL simulations for testing the control software of the ECU dedicated to electromechanical powertrain components. The propulsion system, with its electrical and mechanical components and its interactions with the management transportation system, represents an interesting case-study

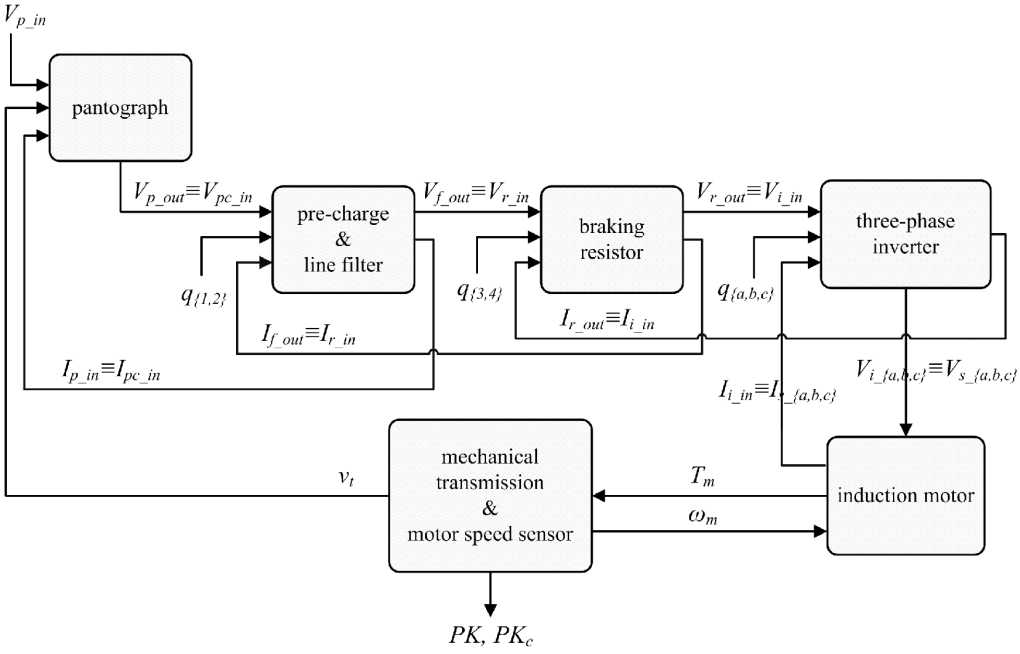
for motivating the usefulness of real-time HIL simulations for railway. On the electrical side, in a modern railway vehicle the electrical and electronic components (filters, supercapacitors, batteries, power converters, switching devices) must operate into architectures characterized by energetic operating conditions of increasing complexity and flexibility. That motivates the adoption of a realistic real-time simulation tool which could be used, for instance, to analyze the system response under specific faults and to test fault-tolerance techniques (Abul Masrur, et al., 2010; Liu, et al., 2009; Facchinetti, et al., 2009). On the mechanical side, the increasing speed of modern trains have made more critical, with increasingly negative economical and technical potential effects, the so-called slipping-sliding behavior in the wheel-rail interaction. In this case, the HIL simulations could be used to test and to analyze the effectiveness of anti-slipping strategies and vehicle absolute position estimation techniques (Bonta, et al., 2006).

In order to ensure the fidelity and to allow the practical exploitation of HIL simulations a fundamental preliminary step consists in constructing "good" models of the system to which the ECU must be interfaced (Busco, et al., 2003). The model fidelity must match its intended use. For instance, if the final target is to test the diagnostic capability of the control unit in the presence of electronic switches faults, the averaged models are not suitable and hybrid power converters models must be used instead; or, in order to verify estimation techniques of the vehicle absolute position, it is fundamental to provide a detailed model of the speed sensor and a compact slipping model for the wheel-rail interaction. More in general, the design of a HIL system aimed at supporting the verification, the validation and the regression testing of the propulsion control software needs as prerequisite the definition of suitable models of the major propulsion chain components: pantograph and its interaction with the line, filters and power converters, auxiliaries, traction motor, bogie, wheels and their interaction with rails. The first part of the chapter will be dedicated to the presentation of the simple enough models of such elements by conducting the reader to learn by examples the typical modular approach needed to construct models for HIL applications

The real-time characteristic of HIL simulations introduces some issues both on the software and hardware parts of the HIL system. A typical hardware solution for allowing the real-time integration of the mathematical models consists of parallel processor units. The major drawback of this solution is on the delays introduced by the needed communications among the different parallel processors. An interesting alternative, which mitigates the problem, is the use of processors together with field programmable gate array (FPGA) boards (Matar and Iravani, 2010; Ren, et al., 2011). The constraint on the maximum computation time for the simulation imposes on this hardware architecture some hierarchies also into the models. Indeed the model of each vehicle component should be divided into "fast" and "slow" parts, the former integrated through FPGA while the latter by using central processing units (CPUs). Such aspects will be presented in the second part of the chapter by analyzing a typical architecture of a HIL system for railway applications and the modeling integrations issues related to such architecture.

The advantages of the real-time HIL simulations for the design and testing of railway control software will be demonstrated in the last part of the chapter by presenting some experimental results based on a real railway vehicle operating under critical conditions. In particular, the set-up includes: electronic (traction) control units, models for singular perturbed electromechanical components and sensors, real-time target computer, real or simulated loads, host computer with communications link to the target computer and diagnostic link to the ECU, adapters needed to interface the simulator to the control unit, a graphical user interface application to download

Figure 1. Block scheme of a typical train propulsion system



and control the real-time process. The case studies that will be considered for the numerical tests are: the disconnection of the phase of the induction motor, the pantograph missing contact and the wheel-rail slipping phenomenon.

LOCOMOTIVE SIMULATOR MODEL

The core of a HIL system is the modeling part, i.e., the mathematical models emulating the real equipment under control. From our HIL perspective the model consists of any part of the train traction system which should provide the inputs to the ECU and should react to the outputs of the ECU. An important question to be addressed is the fidelity level of the models to be used. In general, it is not possible to say a priori whether a model satisfies or not the required fidelity with respect to the real system behavior. Indeed, the model fidelity depends on the specific application and on the operating conditions under test. Typically, HIL systems are used for testing the diagnostic capability or the control capability of an ECU under critical operating conditions. In our case the control unit under test is the so called traction control unit (TCU), which governs the electrical, electronic and mechanical components devoted to the train propulsion system. Figure 1 shows a simplified typical configuration, decomposed into the main elements of our interest:

- The pantograph connected to an ideal overhead line with continuous voltage;
- The pre-charge circuit and the line filter;
- The braking resistor with the corresponding enabling switch, i.e., the so-called braking chopper;
- The three-phase voltage source inverter;
- The induction motor;
- The mechanical transmission system, including its interactions with the rail.

Figure 1 represents not only how the components are physically connected, but also a conceptual scheme on how the dynamic models of the

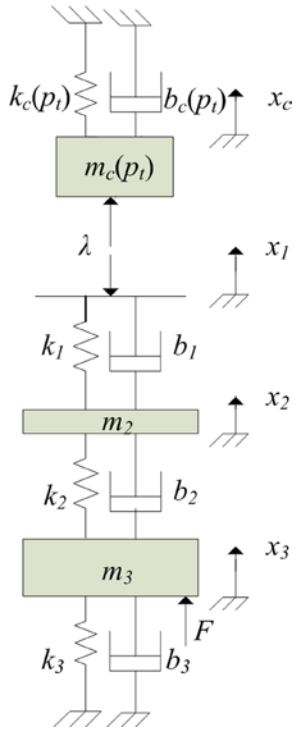
different components are mathematically coupled in order to obtain the model of the entire system. Indeed the different models are not independent and their mathematical interaction is obtained by means of a suitable choice of inputs and outputs of each block. In the section on real-time simulation techniques it will be shown that this modular modeling choice can be also exploited in order to obtain a more efficient numerical integration of the system dynamic equations.

This section is dedicated to the presentation of the powertrain components models, suitably simplified to compromise on simulations fidelity and fast numerical integration. All electrical sensors are modeled by means of constant gains except for the rotor speed sensor for which a more detailed model is required. The model of each subsystem contains also some internal state variables, featured in order to simulate specific critical operating conditions with potentially dangerous effects, e.g., a temporary pantograph disconnection from the overhead line, a fault in one of the motor phases, the slipping wheels phenomenon.

From the Catenary to the Locomotive through the Pantograph

The pantograph connects the overhead line, here represented with a mechanical catenary and an electrical ideal direct current (DC) source, to the electrical train propulsion systems. Because of the mechanical oscillations of the pantograph-catenary system during the train trip, the quality of the current absorbed from the overhead line can dramatically decrease. In particular, as the train speed increases, usually the oscillation amplitude and the frequency increase, causing a more likely partial or complete missing contact between the pantograph and the catenary (Allotta, et al, 2005). The main causes of missing contact between the pantograph and the catenary are the swings of the catenary far from the pylon, due to the aerodynamic force induced by the rapid transit

Figure 2. Equivalent scheme of a catenary-pantograph interaction



of the train. Moreover, another common cause of missing contact is the reduction of the adherence force for a fault in the pantograph actuator. A dramatic consequence of missing contacts is the pantograph arcing, which is more predominant at higher speeds, increasing loads and in cold weather conditions (Midya, et al., 2009).

A three-degree of freedom model of the pantograph and catenary dynamics, in terms of lumped masses, springs and dumpers is considered (Allotta, et al, 2005) and a corresponding scheme is represented in Figure 2. The mechanical parameters of the catenary $m_c(p_t)$, $b_c(p_t)$ and $k_c(p_t)$ are periodic with respect to the train position p_t along each span, instead the symmetric pantograph is modeled by means of a linear lumped-parameters system having as input the (usually constant) pre-load force F. Consider as state variables, indicated with the sym-

Box 1.

$$\begin{cases} m_c(p_t)\dfrac{d^2x_c}{dt^2} + b_c(p_t)\dfrac{dx_c}{dt} + k_c(p_t)x_c = \lambda, \\ \dfrac{dp_t}{dt} = v_t, \\ k_1(x_2 - x_1) + b_1\left(\dfrac{dx_2}{dt} - \dfrac{dx_1}{dt}\right) = \lambda, \\ m_2\dfrac{d^2x_2}{dt^2} = k_2(x_3 - x_2) + b_2\left(\dfrac{dx_3}{dt} - \dfrac{dx_2}{dt}\right) - \lambda, \\ m_3\dfrac{d^2x_3}{dt^2} = F - k_2(x_3 - x_2) - b_2\left(\dfrac{dx_3}{dt} - \dfrac{dx_2}{dt}\right) - k_3x_3 - b_3\dfrac{dx_3}{dt}, \end{cases} \quad (1)$$

bol x, the equivalent masses positions, which are assumed to be zero at rest, and the catenary position x_c. A further input is the train velocity v_t. The dynamics of the eighth order system can be represented by the following set of nonlinear differential equations in Box 1. where λ is the nonnegative contact force possibly connecting catenary part of the model with thepantograph one.

The train position is reset to zero whenever the train reaches a distance multiple of the span length L. In particular, the dependence of the catenary parameters on the train position can be expressed

$$\begin{aligned} m_c(p_t) &= m_{c0} - m_{c1}\cos\left(\frac{2\pi}{L}p_t\right) + m_{c2}\cos\left(\frac{4\pi}{L}p_t\right), \\ b_c(p_t) &= b_{c0} + b_{c1}\cos\left(\frac{2\pi}{L}p_t\right) + b_{c2}\cos\left(\frac{4\pi}{L}p_t\right), \\ k_c(p_t) &= k_{c0} + k_{c1}\cos\left(\frac{2\pi}{L}p_t\right) + k_{c2}\cos\left(\frac{4\pi}{L}p_t\right). \end{aligned} \quad (2)$$

In ordinary operating conditions the pantograph is in full contact with the catenary, which implies x_1=x_c. In the presence of missing contact the contact force is zero. More in general, the contact force λ can be written as

$$\lambda = \max\left\{k_1(x_2 - x_c) + b_1\left(\frac{dx_2}{dt} - \frac{dx_c}{dt}\right),\ 0\right\}. \quad (3)$$

In order to connect the mechanical equations (1) with an equivalent electrical model representing the arcing phenomenon, it is necessary to define a suitable output variable. Consider as inputs of the electrical model the overhead line voltage V_{p_in} and the overhead line current I_{p_in} and as output the voltage V_{p_out} which will be applied to the pre-charge circuit, see Figure1. Then one can write

$$V_{p\_out} = V_{p\_in} - R_{arc}(x_c - x_1)\, I_{p\_in} \quad (4)$$

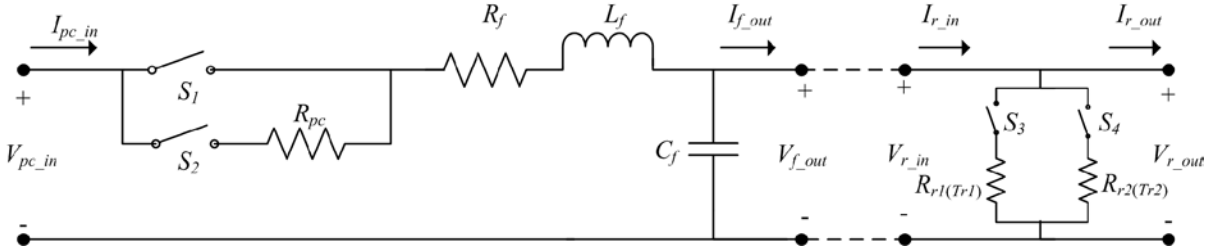
where R_{arc} is the equivalent arc resistance which suitably depends on the arc length x_c-x_1.

We are now ready to move to the models of the electrical components of the locomotive powertrain.

Power Electronic Supply System

In this section, we present the models of the main components of a typical power electronic system for electric locomotives with a DC

Figure 3. Equivalent circuit of the pre-charge switched circuit with line filter and braking chopper



overhead line: the pre-charge circuit and the line filter, the chopping braking resistor with its thermal dependence, the voltage source inverter with dead time, the three phase induction motor with possible phase disconnection. Obviously, many other architectures of the power electronics propulsion system are possible, but the considered topology is of interest because on one side it is widely used for different trains and, on the other side, it allows to present the modular modeling procedure which can be similarly applied to other electrical and electronic configurations.

Pre-charge and Line Filter

An equivalent electrical circuit of the pre-charge circuit and line filter is shown in Figure 3. The pre-charge circuit is used for limiting the inrush current flowing into the filter capacitor during power-up. That allows to reduce and to limit the power-up stress and to increase the components life. The circuit is realized through a series of controlled electronic switches (typically bipolar junction transistors or insulated gate bipolar transistors), S_1 and S_2, and an appropriate resistance R_{pc}, which is switched out upon completion of the pre-charging phase, that is when the output voltage V_{f_out} is close to its nominal value. The line filter is used to attenuate the possible electromagnetic interference between the line and the equipment, and it is represented as a classical R_f, L_f and C_f electrical circuit.

A classical approach for representing the switching behavior of power electronics systems consists of idealizing the electronic switch as a two states device: when the switch is ON it is represented as a short cut, whereas the switch in the OFF state is modeled as an open circuit (Vasca, et al., 2009). The integration of the switch behavior inside the dynamic model of the electrical system can be obtained by introducing the so-called switching function q_i for the switch i, with $i \epsilon \{1,2\}$:

$$q_i = \begin{cases} 1 & if \text{ switch } i \text{ ON;} \\ 0 & if \text{ switch } i \text{ OFF.} \end{cases} \tag{5}$$

The switching function can be interpreted as the control variable which determines whether the switch is conducting (ON) or blocking (OFF). Consider as inputs of our dynamic model the pre-charge filter input voltage V_{pc_in} and the line filter output current I_{f_out}, and as outputs the line filter output voltage V_{f_out} and the pre-charge circuit input current I_{pc_in}. By assuming that both switches are never simultaneously ON and by applying the Kirchhoff laws, the system can be described by the following equations in Box 2.

Note that if both switches are OFF the second equation in (6) reduces to the algebraic constraint that the input current must be zero.

Box 2.

$$\begin{cases} C_f \dfrac{dV_{f\_out}}{dt} = (q_1 + q_2) I_{pc\_in} - I_{f\_out}, \\ (q_1 + q_2) L_f \dfrac{I_{pc\_in}}{dt} = (V_{f\_in} - V_{f\_out} - (R_f + q_2 R_{pc}) I_{pc\_in})(q_1 + q_2) + (1 - q_1)(1 - q_2) I_{pc\_in}. \end{cases} \quad (6)$$

Braking Chopper

When the speed of the inverter-controlled induction motor decreases, the motor acts as a generator feeding back energy to the supply system. This situation may determine dangerous working conditions for the electrical and electronic components. A typical solution for non regenerative overhead lines consists of dissipating the electrical power into the so called braking resistor, which is specifically designed to absorb energy and dissipate it into heating. In particular, when the output voltage of the filter V_{f_out} exceeds a specified threshold, an electrical switch connects the line filter with the braking.

Figure 3 shows the equivalent circuit with two braking resistors and the corresponding enabling switches S_3 and S_4. They represent the so called braking choppers, while $R_{r1}(T_{r1})$ and $R_{r2}(T_{r2})$ are the temperature dependent resistances. In particular, by modeling the thermal behavior of the resistances it is possible to monitor the temperature evolution of the braking resistor and to check critical values exceeding.

In order to write the system equations, it can be useful to use the switching function definition (5) with $i \in \{3,4\}$. The inputs of the model are: the switching functions q_i, the voltage V_{r_in} which will be equal to the output voltage of the line filter (see Figure1) and the current I_{r_out} which will be determined from the inverter model. The outputs of the model are: the voltage V_{r_out} which represents the voltage source for the inverter and the current I_{r_in} which is the load current for the filter (I_{f_out}). Then, the (static part of the) model can be written as seen in

$$\begin{cases} V_{r\_out} = V_{r\_in}, \\ I_{r\_in} = \dfrac{V_{r\_in}}{R_{r1}(T_{r1})} q_3 + \dfrac{V_{r\_in}}{R_{r2}(T_{r2})} q_4 + I_{r\_out}. \end{cases} \quad (7)$$

A linear dependence of the resistance on the temperature is assumed:

$$R_{ri} = R_{ri0}\left[1 + \alpha(T_{ri} - T_0)\right] \quad (8)$$

where T_0 is the nominal temperature, R_{ri0} the corresponding nominal resistance, and α is the temperature coefficient of resistance. Finally, in order to complete the braking resistor model we associate to the static equations (7)–(8) the following differential equation describing the time evolution of the temperature

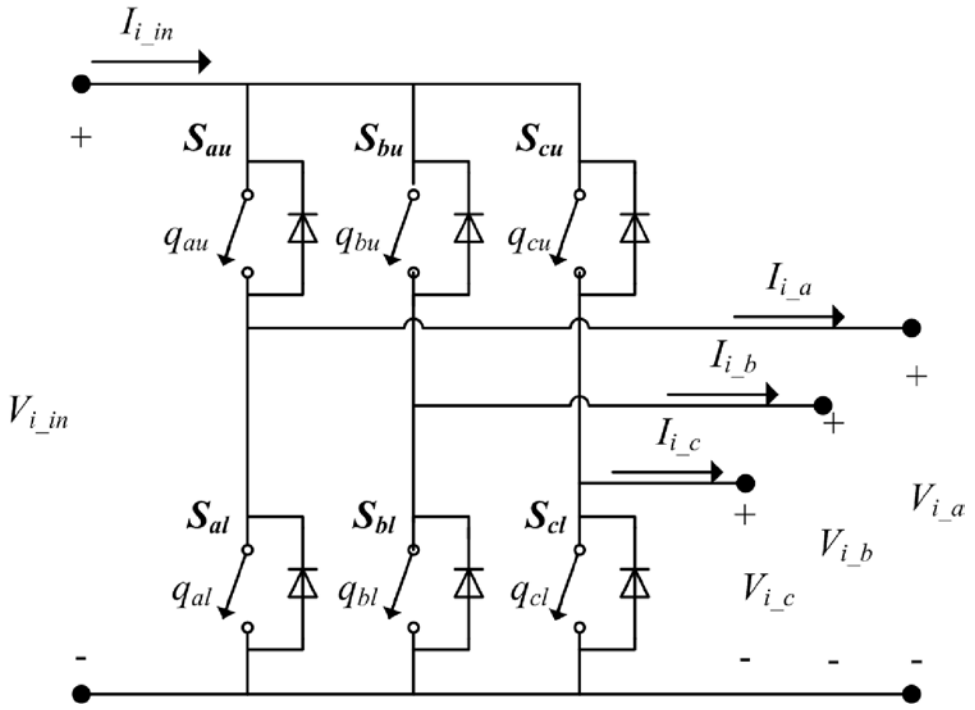
$$c_p \frac{dT_{ri}}{dt} = R_{ri} I_{ri}^2 - \beta(T_{ri} - T_{env}) \quad (9)$$

with β the thermal dispersion coefficient and c_p the thermal absorption coefficient. The model (9) is clearly nonlinear because the resistor current I_{ri} depends on the inverse of the thermal-dependent resistance, see (7) and (8).

Three-phase Voltage Source Inverter

A classical voltage source inverter is considered for generating the three phase voltages supplying the induction motor (Mohan, et al., 2003; Di

Figure 4. Equivalent circuit of the three-phase voltage source inverter



Pietro, et al., 2010). The structure of the inverter is shown in Figure 4, where V_{i_a}, V_{i_b}, V_{i_c} are the voltages applied to the motor stator windings, V_{i_in} is the inverter input voltage (equals to the braking chopper voltage V_{r_out}) and where S_{ju} and S_{jl} with $j\epsilon\{a,b,c\}$ represent the six switches of the inverter. The two switches on the same leg, i.e., S_{ju} and S_{jl}, cannot be ON simultaneously because a short circuit across the DC link voltage source would be produced. Therefore the upper and lower switches of the same leg j, respectively S_{ju} and S_{jl}, are driven with two complementary pulsed signals.

Analogously to (5) it is useful to define a switching function q_j for the leg j, with $j\epsilon\{a,b,c\}$:

$$q_j = \begin{cases} 1 & if\ \{S_{ju}\ \text{ON}\ \wedge\ S_{jl}\ \text{OFF}\} \vee \{S_{ju}\ \text{OFF}\ \wedge\ S_{jl}\ \text{OFF} \wedge I_{i\_j} \leq 0\} \\ 0 & if\ \{S_{ju}\ \text{OFF}\ \wedge\ S_{jl}\ \text{ON}\} \vee \{S_{ju}\ \text{OFF}\ \wedge\ S_{jl}\ \text{OFF} \wedge I_{i\_j} > 0\}. \end{cases} \quad (10)$$

So as mentioned above, we assume that it is not allowed that the two switches of the same leg are simultaneously ON. On the other hand, the presence of the dead-time allows the two switches of the same leg to be OFF at the same time. In such situation the configuration of the converter is determined by the conducting diodes in parallel to the switches. In particular, by considering the typical idealized diode characteristic (Kassakian, et al., 1991), it will conduct the diode in parallel with the upper switch (lower switch, respectively) if the corresponding phase current will be negative (positive, respectively).

So as reported in Figure 1, the inputs of the inverter model are: the switching functions q_j with $j\epsilon\{a,b,c\}$, the phase currents I_{i_j} which comes from the induction motor model and the voltage V_{i_in} which is the output voltage of the braking resistor model. The outputs of the inverter model are the three phase voltages V_{i_j} and the inverter current I_{i_in}. So, by introducing the following vector notation in Box 3.the (static) model of the inverter can be simply written as

Box 3.

$$V_{i\_out} = \begin{bmatrix} V_{i\_a} & V_{i\_b} & V_{i\_c} \end{bmatrix}^T ;\ I_{i\_out} = \begin{bmatrix} I_{i\_a} & I_{i\_b} & I_{i\_c} \end{bmatrix}^T ;\ q = \begin{bmatrix} q_a & q_b & q_c \end{bmatrix}^T \quad (11)$$

$$\begin{cases} V_{i\_out} = qV_{i\_in}, \\ I_{i\_in} = q^T I_{i\_out}. \end{cases} \quad (12)$$

As it will be shown in the section on the HIL setup, the implementation of the switching functions in the HIL system requires the use of fast computing boards (FPGA) in order to detect with sufficient accuracy the time instant of the switching edges.

Induction Motor with Phase Disconnection

The three phase induction motor is the most commonly used electric motor in railway applications. The well known dynamic model of the induction motor can be written by using as state variables the stator currents, the rotor fluxes and the rotor speed (Bose, 2006). In this section we present the model of the electrical part of the motor, including the calculation of the electromagnetic torque. Instead, the mechanical equation representing the rotor speed time evolution will be presented in the next section because this part is more strictly related to the mechanical shafts equations.

By considering a balanced asynchronous motor, the electrical dynamic equations written in the stationary *d-q* reference frame are

$$\left(1 - \frac{M^2}{L_s L_r}\right) L_s \frac{d\bar{I}_s}{dt} = -\left(R_s + R_r \frac{M^2}{L_r^2}\right)\bar{I}_s + \frac{M}{L_r}\begin{bmatrix} R_r/L_r & n_p\omega_m \\ -n_p\omega_m & R_r/L_r \end{bmatrix}\bar{\Phi}_r + \bar{V}_s, \quad (13)$$

$$\frac{d\bar{\Phi}_r}{dt} = R_r \frac{M}{L_r}\bar{I}_s + \frac{1}{L_r}\begin{bmatrix} -R_r & -n_p\omega_m M \\ n_p\omega_m M & -R_r \end{bmatrix}\bar{\Phi}_r \quad (14)$$

where ω_m is the rotor speed in mechanical radians, n_p is the number of poles pairs, L_s and L_r are the stator and rotor inductances, R_s and R_r are the stator and rotor resistances, M is the mutual inductance, and the rotor flux vector $\bar{\Phi}_r$, the stator current vector $\bar{I}_s$ and the stator voltage vector $\bar{V}_s$ are given by

$$\bar{\Phi}_r = \begin{bmatrix} \phi_d & \phi_q \end{bmatrix}^T, \quad \bar{I}_s = \begin{bmatrix} \iota_d & \iota_q \end{bmatrix}^T, \quad \bar{V}_s = \begin{bmatrix} v_d & v_q \end{bmatrix}^T. \quad (15)$$

The model (13)-(15) clearly shows that only two components of the three phase currents (and voltages and fluxes) are needed in order to represent the dynamic behavior of the motor. This is due to the equilibrium condition among the three components of the electrical variables. Indeed the equations in the stationary *d-q* reference frame are obtained by means of the well known Park transformation, which, by considering as an example the stator currents, can be written as

$$\bar{I}_s = \begin{bmatrix} \iota_d \\ \iota_q \end{bmatrix} = P_{dq}\begin{bmatrix} I_{s\_a} \\ I_{s\_b} \\ I_{s\_c} \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} I_{s\_a} \\ I_{s\_b} \\ I_{s\_c} \end{bmatrix} \quad (16)$$

and analogously for the rotor fluxes and the stator voltages. The inverse transformation can be written as

$$\begin{bmatrix} I_{s\_a} \\ I_{s\_b} \\ I_{s\_c} \end{bmatrix} = P_{abc}\bar{I}_s = P_{abc}\begin{bmatrix} \iota_d \\ \iota_q \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1/2 & \sqrt{3}/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} \iota_d \\ \iota_q \end{bmatrix} \quad (17)$$

Box 4.

$$\left(1-\frac{M^2}{L_s L_r}\right)L_s\frac{dI_{s\_a}}{dt}+\left(R_s+R_r\frac{M^2}{L_r^2}\right)I_{s\_a}+\delta(\phi_d,\phi_q,\omega_m)=V_{s\_a} \tag{19}$$

Finally, the coupling between the induction motor and the mechanical transmission is obtained through the electromagnetic torque T_m, see Figure 1, which can be written as

$$T_m=\frac{3}{2}n_p\frac{M}{L_r}(\phi_d\iota_q-\phi_q\iota_d). \tag{18}$$

The stator of the induction motor is connected to the three terminal wires supplying the motor. When one of these terminal wires is disconnected from the stator a so called open terminal fault occurs. This is a very dangerous condition for the entire drive and therefore it is important to simulate such a situation using the real-time HIL system. To this aim the induction motor model must be suitably modified when such fault occurs. More precisely, the model with phase disconnection is derived by the previously described dynamic model of the induction motor by properly changing the stator voltage of the faulty winding. When this type of fault occurs, the corresponding current rapidly goes to zero. The fault simulation is realized in two steps: in the first step it is generated a phase voltage which forces the current to reach zero with a very fast dynamic, whereas in the second step it is generated a phase voltage such that the current remains identically zero. In order to obtain the needed expressions of the voltages in the two step we can directly refer to (13)-(14), which is still valid because no fault is assumed to occur inside the motor. Without loss of generality, we assume that the phase which is subject to the disconnection is the phase *a*. From (16) and (17) it is clear that $I_{s\_a}=\iota_d$, and also $V_{s\_a}=v_d$. Then from (13) we get (see Box 4).where

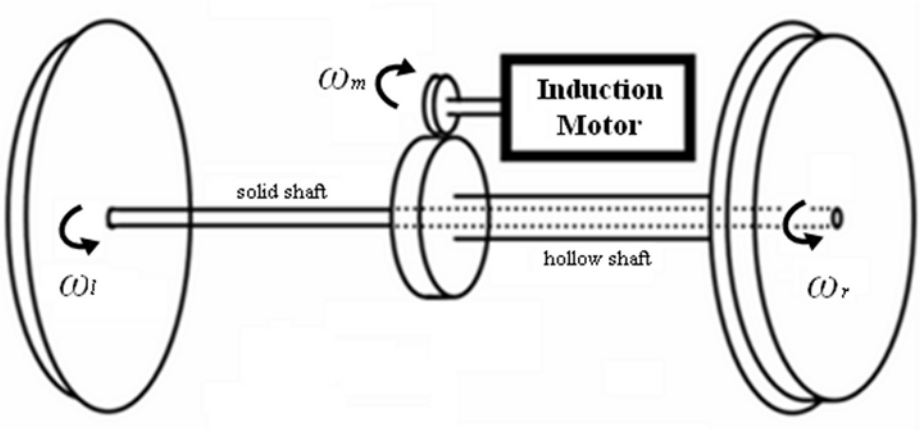
$$\delta(\phi_d,\phi_q,\omega_m)=-\frac{R_r M}{L_r^2}\phi_d-n_p\omega_m\frac{M}{L_r}\phi_q \tag{20}$$

For the design of the phase voltage $V_{s\_a}\neq V_{i\_a}$ in the first step the term δ in (20) is interpreted as a disturbance. Then if the phase current I_{s_a} is positive (negative, respectively) it is generated a large negative (positive) phase voltage V_{s_a} letting the current to reach zero fast. Then, when the phase current becomes zero, in order to let the phase current to remain zero, the phase voltage V_{s_a} is obtained from (19) by assuming the current and its derivative to be identically zero, i.e. $V_{s\_a}=\delta$. The other two phase voltages are generated according to (12), i.e. $V_{s\_b}=V_{i\_b}$ and $V_{s\_c}=V_{i\_c}$.

Mechanical Transmission and Pick-Up Model

In this section we present the models used in the HIL system to simulate the mechanical transmission of the locomotive. The mechanical model is simplified to the aerodynamic motion resistances and the track of the train is considered flat without curves or slopes. This hypothesis allows to neglect all the transversal forces and to consider only longitudinal ones. This model might appear oversimplified for a detailed analysis of the train trip, but, similarly to the electronic components models, it allows to present the proposed modular

Figure 5. Equivalent scheme of the mechanical transmission system from motor to wheels



modeling procedure. Furthermore, for our specific simulation scenarios, the models presented below represent a good compromise between the model fidelity and the low complexity required to satisfy the constraints on the real-time sampling rate imposed by the hardware adopted for the HIL set-up.

The motor is connected to the wheels by means of a transmission system. A pair of wheels is rigidly fixed to an axle to form a wheelset. A gearbox is connected to the motor shaft and to the axle which consists of two shafts, one solid and the other hollow, so transmitting the torque to the wheels. An equivalent scheme is shown in Figure 5.

The wheelset is linked to the bogie by means of bearings. Usually two wheelsets are mounted in each bogie: a driving wheelset which is connected to the motor, and a trailer wheelset. In order to get a model of the driving wheelset we assume the solid and hollow shafts to be flexible with constant concentrated parameters characterizing stiffness and damping. By simplicity, justified by the high rigidity of vehicle suspension, the effects of the dynamical transfer load between the axles are neglected. So, the dynamics of the driving wheelset can be described by means of the following differential equations

$$\begin{cases} J_m \dfrac{d\omega_m}{dt} = T_m - \dfrac{k_r}{\rho}\left(\dfrac{\vartheta_m}{\rho} - \vartheta_r\right) - \beta_m \omega_m, \\ J_r \dfrac{d\omega_r}{dt} = k_r\left(\dfrac{\vartheta_m}{\rho} - \vartheta_r\right) - k_l\left(\vartheta_r - \vartheta_l\right) - r_r F_r - \beta_r \omega_r, \\ J_l \dfrac{d\omega_l}{dt} = k_l\left(\vartheta_r - \vartheta_l\right) - r_l F_l - \beta_l \omega_l \end{cases} \tag{21}$$

where the subscripts m, r, l and w are used to indicate motor shaft, right wheel, left wheel and wheelset, respectively, while J are inertias, ω speeds, ϑ angles, β damping coefficients, r wheel radius, F adherence forces, T torques, k stiffness coefficients and ρ is the gearbox ratio.

The locomotive motion is determined by a complex interaction between wheels and rails. On the wheelset side such interaction is represented by the adherence forces F_r and F_l which appear in (21) as load forces. Instead on the point of view of the locomotive motion the adherence forces determine the longitudinal movement. We assume the wheelset to be rigidly connected to the bogie, so as the bogie to the locomotive, an uniform distribution of the locomotive and bogie weights and an one-dimensional motion characterized only by the longitudinal direction. Say M_{loc} the locomotive mass, M_b the bogie mass, v_w the wheelset longitudinal speed, F_L the load force,

n_b the number of bogies per locomotive and n_a the number of driving wheelset per bogie. Then we can write the following dynamic equation representing the Newton law

$$\frac{1}{n_a}\left(\frac{M_{loc}}{n_b}+M_b\right)\frac{dv_w}{dt}=F_r+F_l-\beta_w v_w-F_L. \tag{22}$$

From (30)-(31) it is clear that a detailed characterization of the wheel-rail contact mechanism represented by the adherence forces is needed in order to permit a correct analysis of the corresponding dynamic behavior. This is a classical problem in the railway literature (Kalker, 1991). The simplest approach to tackle the problem consists in considering a pointwise contact and a suitable dependence of the adherence forces on the so called slipping velocity, i.e., the difference between the longitudinal wheelset velocity and the circumferential velocity of the wheel. As it is well known from classical mechanics as soon as an accelerating or braking torque is applied to the wheels a slipping will occur.

Let us define the slip of the left and right wheel s_l and s_r respectively as follows:

$$s_l=\frac{r_l\omega_l-v_w}{r_l\omega_l}, \quad s_r=\frac{r_r\omega_r-v_w}{r_r\omega_r}. \tag{23}$$

The adherence forces can be expressed as a fraction of the normal force exerted on the specific wheel. By assuming a uniform weight distribution one can write

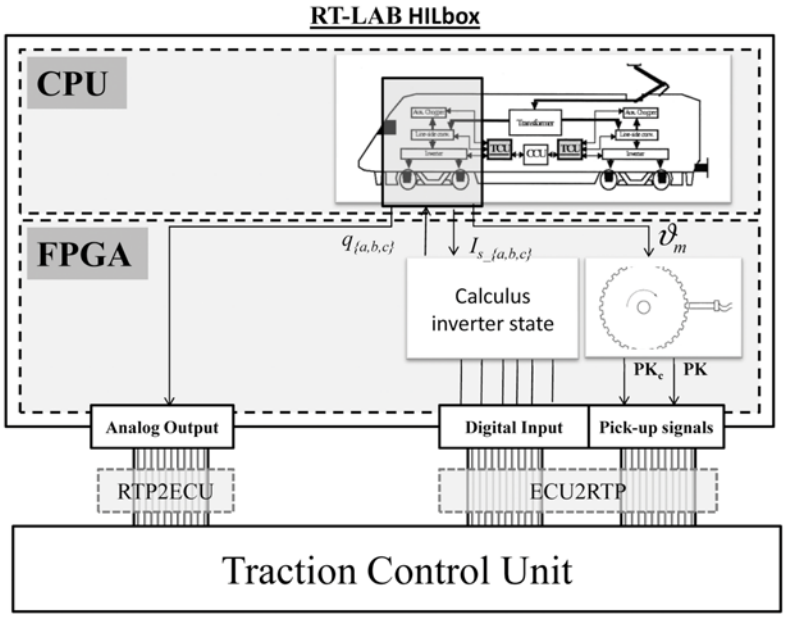
$$\begin{aligned} F_l&=\frac{1}{n_c}\left(M_b+\frac{M_{loc}}{2}\right)gf_a(s_l),\\ F_r&=\frac{1}{n_c}\left(M_b+\frac{M_{loc}}{2}\right)gf_a(s_r)\end{aligned} \tag{24}$$

where n_c is the total number of wheels per bogie, g is the gravity acceleration and f_a is the so called adherence coefficient which depends on the slip of the corresponding wheel. With respect to the profile of the adherence coefficient, the form and the initial slope for wet and dry conditions are different. The model presented above is sufficiently simple to be integrated with the models of the electrical components of the powertrain. The price to pay is to neglect several details of the wheel-rail contacts which can be analyzed with more detailed HIL models (Zhou, et al., 2010).

Before completing this section it is important to clarify some aspects related to the modeling of the speed sensor. This is an important issue because many rail vehicle control and monitoring systems depend on a reliable and precise speed measurements. For the HIL system under investigation a model of the phonic wheel with speed sensor has been constructed (Nyce, 2004). The phonic wheel consists of a toothed wheel of soft iron. The phonic wheel works in association with the pick-up, a magnetic sensor that detects a sequence of "voids" and "solids" and consequently generates a consistent square wave electrical signal. The pick-up, which is a bar electromagnet, is fixed with one pole facing the teeth of the wheel and generates a sequence of electrical impulses with angle interval $2\pi/n_{teeth}$ where n_{teeth} is the number of teeth of the phonic wheel. The computation of the time interval between two successive impulses allows the TCU to compute the motor speed. To this aim the following two signals are sent to the TCU

$$\begin{aligned} PK&=\begin{cases}1 & \text{if} \quad \vartheta_m \bmod \dfrac{2\pi}{n_{teeth}} \geq \dfrac{\pi}{n_{teeth}}\\ 0 & \text{if} \quad \vartheta_m \bmod \dfrac{2\pi}{n_{teeth}} < \dfrac{\pi}{n_{teeth}}\end{cases}\\ PK_c&=\begin{cases}1 & \text{if} \quad \left(\vartheta_m+\dfrac{\pi}{2n_{teeth}}\right) \bmod \dfrac{2\pi}{n_{teeth}} \geq \dfrac{\pi}{n_{teeth}}\\ 0 & \text{if} \quad \left(\vartheta_m+\dfrac{\pi}{2n_{teeth}}\right) \bmod \dfrac{2\pi}{n_{teeth}} < \dfrac{\pi}{n_{teeth}}\end{cases}\end{aligned} \tag{25}$$

Figure 6. Real-time HIL set-up general scheme



In particular the binary signal *PK* is enough to compute the angular speed, whereas a suitable combination of *PK* and PK_c allows to determine the direction of rotation of the motor shaft.

HIL SET-UP

The models presented above have been used for realizing the real-time HIL simulations. A general scheme of the HIL set-up used to simulate the case study of interest is shown in Figure 6. It includes four main components:

- A real-time simulator of the equipment under control, implemented with the RT-LAB HIL box, where the models of the electromechanical components of the locomotive are numerically integrated;
- An electronic control unit (ECU) corresponding to the hardware under test, which in our case is the traction control unit (TCU);
- Two adapters (ECU2RTP and RTP2ECU) which convert signals from the ECU to the RT-LAB HIL box and vice-versa.

The core of the HIL set-up is the real-time simulator. Personal computers with common operating systems, although of high performance, are not suitable to serve real-time application requests. Indeed, several real-time apparatus suitable to realize HIL simulations are commercially available. For our purposes, RT-LAB HILbox has been used as the real-time hardware platform. RT-LAB® is a distributed real-time platform, provided by Opal-RT Technologies (http://www.opal-rt.com/), that enables the distributed simulation of complex electromechanical devices, power systems, and controllers on multi CPU-FPGA targets. RT-LAB allows the user to readily convert models written in the environment MATLAB®/Simulink® by Mathworks (http://www.mathworks.com/), via Real-Time Workshop, and then to conduct real-time simulation with hardware-in-the-loop. It builds parallel tasks

from the original Simulink model and run them on each CPU of the multi-CPU computer, where data are exchanged through shared-memory. In particular, the RT-LAB HILbox simulator includes a multi-core processing unit, where a single core is dedicated to execute the code for the communication protocol (Ethernet 100Mb/s UDP/IP standard 100baseTx with TIA/EIA-568-B) with an external command station. In order to speed-up the real-time simulations the command station can generate only asynchronous commands for the RT-LAB HILbox and then the specific communication protocol is not crucial. Possible synchronous events which influence the real-time simulations can be generated by including specific modules inside the models simulated by the RT-LAB HILbox. The other cores are dedicated to run the system models. Furthermore, the simulator is provided of more than one reconfigurable FPGA cards, used both for input/output (I/O) signals conditioning and pre-post processing, as well to simulate special parts of the system, whose require fast sample time with respect to the minimum sample time of CPU. In this case, a Simulink blockset, called Xilinx System Generator, is used to design FPGA-based models and provide automatic code generation from MATLAB/Simulink.

We can now make some considerations which relate the HIL set-up architecture of Figure 6 with the mathematical models of the electromechanical components of Figure 1. The analog signals from the RT-LAB HILbox to the TCU are the line, braking chopper and phase motor currents and the line and DC link voltages. The digital signals from the RT-LAB HILbox to the TCU are the pick-up signals, while the digital signals from the TCU to the RT-LAB HILbox are the six pulse width modulation (PWM) signals, one for each switch of the inverter, as well the switches' command signals for the resistor and pre-charge circuit.

Inside the RT-LAB HILbox the partitioning of the model integration between CPU and FPGA boards is quite useful to simulate the part of the system which present a natural "stiff" behavior, requiring very small time steps or variable-step solvers to achieve convergence and accuracy, which are prohibitive for real-time simulations with classical CPU-based HIL simulators. The equations representative of the whole system in Figure 1, except for the speed sensor model and part of the inverter, are solved in CPU, while the I/O operations, the processing of pick-up signals, as well the determination of the state of each inverter leg through the generation of the signals $q_{\{a,b,c\}}$, see (10), are executed on the FPGA board. In this way the CPU-based model runs at a 25 μs rate, while the FPGA-based model run in parallel with 10 ns of time step. Furthermore, both CPU and FPGA communicate synchronously through Opal-RT Signal-Wire link.

The pick-up signals (25) of the speed sensor model are generated in FPGA because the updating rate required by the TCU is less than the minimum CPU time step. Due to the different sampling times between CPU and FPGA, an interpolation method is used to make consistent the signal exchanged between the two boards.

The generation of the switching functions $q_{\{a,b,c\}}$ (10) in FPGA is motivated by the fast switches commutations. In particular, the use of FPGA instead of CPU allows to catch with a good accuracy the switching edges of the PWM signals. From the switches commands received from the TCU, the state of each leg of the inverter is determined according to the following relation

$$q_j = n_j \frac{T_{FPGA}}{T_{CPU}} \tag{26}$$

where T_{FPGA} is the sampling period of the FPGA, T_{CPU} is the sampling period of the CPU, n_j is an integer obtained by means of a counter determining how many times a nonzero signals q_j is received by the FPGA from the TCU within a period T_{CPU}. When both switches of the same leg are open the value of q_j is determined by the considering the sign of the

current, see (10). The switches signals q_j calculated by the FPGA are then sent to the CPU. This approach it is not used for computing the state of the other switches of the system, because a lower accuracy for their switching time detections is required.

TECHNIQUES FOR AN EFFICIENT REAL-TIME SIMULATION

The models of the electromechanical locomotive components and the HIL set-up presented in the previous sections have shown the importance for an efficient numerical integration of the power electronic components models. From the real-time simulation point of view these modules are the most critical part of the entire system because of the interaction between the very fast dynamics typical of the switching electronic devices and the slow dynamics corresponding to the time evolutions of the electrical and mechanical variables. Moreover, the powertrain layout considered above represents only a particular example of a wide spectrum of possible power electronics architectures of modern locomotives, e.g., four-quadrant converters, active filters, power electronics transformers, choppers. The considerations above motivate the presentation in this section of some more general considerations on the integration issues for real-time HIL simulations of power electronic systems. The inverter model (10)-(12) will be considered as the guiding example.

Model Parallelization

The model parallelization can improve the real-time simulation at different levels of abstraction. So as shown in the previous section by considering the inverter model, a possible solution consists of distributing the models between CPU and FPGA boards. The use of FPGA improves the performance of the real-time simulation because of its massively parallel structure.

By considering the CPU, an improvement can be obtained by using the parallel multi-core CPU architecture properly. In particular, in real-time simulations for power electronic systems there are different possibilities to exploit parallelism in order to integrate the differential equations representing the system under test. The presence of multi-core and the need for data exchange among them implies the presence of transmission delays. This is not a problem if the state variables dynamics are much slower than the simulation time step: one can decouple the complete system in several parts and adding a delay to transmit the state from one core (and its task) to the other will not much affect the accuracy of the entire simulation. Another alternative solution is to let correspond the delay between cores to a natural delay of the physical system. For instance, for an HIL simulation of several trains operating on the same electrical transmission line one can associate the model of a train to each core, thus exploiting the slow dynamics of the electrical variables of the transmission line.

The association between the model (or sub-model) and the board on which the model must be integrated is a choice left to the user of the real-time HIL set-up. Another fundamental contribution for an efficient real-time simulation comes from the choice of the numerical integration technique which is usually done by designer of the HIL simulator. In the remaining part of this section the most important solving techniques used in real-time simulation of power electronic systems are presented. Each technique presents both advantages and disadvantages, and a specific choice of one of them depends on several factors, such as the desired accuracy of the results, the available hardware resources, the purposes of the simulation tests.

Explicit Fixed-Causality Solver

A model for real-time simulation usually consists of dynamic equations, such as (1), (6), (9), (13)-(14) and (21), and static equations

or discrete event conditions, such as (3), (5), (10), (25). The causality of the explicit dynamic equations is a straightforward consequence of their time discretization. Instead, static equations may introduce algebraic loops which complicate the solution of the entire model. For instance, consider the inverter model (10)-(12). The switching signals depend on the sign of the phase current, which depends on the specific mode of the inverter, i.e., the converter active topology. We can fix the causality of the equations just introducing a delay in the current which determines the switching signals in (10).

Another useful expedient used in explicit fixed-causality solvers consists of substituting the instantaneous switching functions with their time-average over the sampling period (Pedicini, et al., 2011). This technique often produces accurate results, without any imprecision caused by simulator sampling effects because of the averaging process. However, the model has two important limitations. First, when both switches of the same leg are OFF, i.e., no switching signals are applied to the switches, and the current drops to zero, a real power converter goes into a so-called high-impedance mode. Since the considered model is characterized by a sort of equivalent controlled voltage sources, see (12), this situation is not easy to implement. Some pragmatic solutions have been found to this problem in which a fictitious switch is added in series with the converter output in order to simulate high-impedance effects (Harakawa, et al., 2009). Second, by using this type of solver, due to the averaging process, it is not simple to implement the simulations of a realistic switching device faults.

Implicit Nodal Solver on FPGA

The nodal technique is a classical and trusted solver used mainly for power system simulation (Dommel, 1969). It models switching elements as binary switches with a very high conductance when the switch is in conduction state (ON) and very low one when the switch is not conducting (OFF).

Using Kirchhoff laws, a nodal equation is built from the individual elements and solved at each time step. As an example consider the phase *a* of the inverter in Figure 4. Say $G_{au/l}$ the conductance corresponding to the switch of the upper/lower part of the leg, and assume V_{i_in} and I_{i_a} being known prior to entering the time step. We look for the solution of the V_{i_a} nodal voltage from the algebraic equations

$$\begin{bmatrix} G_{au} + G_{al} & -G_{au} \\ -G_{au} & G_{au} \end{bmatrix} \begin{bmatrix} V_{i\_a} \\ V_{i\_in} \end{bmatrix} = \begin{bmatrix} I_{i\_a} \\ 0 \end{bmatrix} \tag{27}$$

Then

$$(G_{au} + G_{al})V_{i\_a} = G_{au}V_{i\_in} + I_{i\_a} \tag{28}$$

from which V_{i_a} can be obtained. The nodal method described here can of course be extended to simulate capacitive and inductive elements. In this case the differential equation is discretized by trapezoidal or backward-Euler method and incorporated into the formulation in (28). The implicitness comes from the fact that each individual element has unknown current and voltage prior to the iteration that cannot be computed only from the element previous values. Also, when the DC-link is composed of an inductor-capacitor filter instead of a fixed DC voltage source, V_{i_in} becomes an unknown voltage of the nodal system and links the three arms' equations.

The nodal solver method for motor drives is better applied to FPGA computational engines for the reason that interpolation is not required due to the high sampling rate of the switches. Interpolation is notably difficult to implement within a nodal solver. On the other side, the re-

alization of a nodal solver on an FPGA chip can be tricky because of the implementation cost of some basic mathematical function. For example, implementing a division operation on an FPGA chip can be prohibitive and should be avoided. Consequently, to implement the nodal solver on the FPGA, it is often required to pre-compute all the possible inverses of nodal matrix, e.g., the inverse of $G_{au}+G_{al}$ in (28), store them in memory and access them when required during the real-time loop. This, in return, imposes some limits on the number of switches of the inverter device because of the memory requirements (one needs to store 2^N permutations of the nodal matrix, where N is the number of switches of the nodal system). It should be noticed that with the nodal solver approach it is not possible to simulate ideal switches, i.e., electronic devices modeled as short circuit (infinite conductance) when ON and as an open circuit (zero conductance) when OFF.

Partitioning of Nodal Solvers with State-Space Models

An interesting improvement of the implicit nodal solver technique for real-time simulations is represented by the state-space nodal (SSN) solver technique. Similar to the solver used for EMTP® (Electro Magnetic Transient Program, http://www.emtp.org/), state-space nodal solvers are used in commercially available tools for the simulation of power converters, such as SimPowerSystem (http://www.mathworks.com/) and PLECS® (http://www.plexim.com/), and are also used in RT-LAB (http://www.opal-rt.com/product/rt-lab-professional; Dufour, et al., 2011). For simplicity let us assume that the power electronic system in each mode can be represented by means of a set of continuous-time linear time-invariant equations:

$$\begin{cases} \dfrac{dx}{dt} = A^k x + B^k u \\ y = C^k x + D^k u \end{cases} \tag{29}$$

where u is the vector of the inputs of the system (typically voltage and current sources), x is the vector of states of the system (typically inductor currents and capacitor voltages), y is the vector of output of the systems (typically currents, flux and voltages) and k is the switch permutation index (it is an integer ranging from 1 to 2^N, N being the number of binary switches). The use of a SSN solver is complicated by two typical realistic scenarios:

- The power electronic system usually contains a large number of switches and then the system works in an extremely large number of circuit modes;
- The forced-commuted topology of some converters implies that some switches commute instantaneously as a reaction to other switches commutations; the detection of these simulation "impulse events" is a challenge in real-time.

The classical approach to make real-time simulations of systems described with state-space variables is to pre-compute all possible sets of system matrices and store these matrices in memory prior to the real-time loop. However, the method has the major drawback of requiring huge memories to store all sets of matrices when the number of switches increases. SSN solver was designed to solve this problem by introducing the notion of state-space groups that are linked by a nodal matrix. With this approach, the number of switch per groups can be limited to a reasonable number. The key idea of SSN technique is to introduce "nodes" (as in the EMTP nodal method sense) in the system of equations and to use these nodes to introduce a decoupling between the groups of state equations. These groups can

Box 5.

$$x_{n+1} = A_d^k x_n + B_{d1}^k u_n + \begin{bmatrix} B_{d2(in)}^k & B_{d2(no)}^k \end{bmatrix} \begin{bmatrix} u_{n+1(in)} \\ u_{n+1(no)} \end{bmatrix}$$
$$\begin{bmatrix} y_{n+1(in)} \\ y_{n+1(no)} \end{bmatrix} = \begin{bmatrix} C_{d(in)}^k \\ C_{d(no)}^k \end{bmatrix} x_{n+1} + \begin{bmatrix} D_{d(in-in)}^k & D_{d(in-no)}^k \\ D_{d(no-in)}^k & D_{d(no-no)}^k \end{bmatrix} \begin{bmatrix} u_{n+1(in)} \\ u_{n+1(no)} \end{bmatrix} \quad (31)$$

Box 6.

$$y_{n+1} = W^k u_{n+1(no)} + f(x_n, u_n, u_{n+1(in)})$$
with (32 and 33)
$$W^k = C_{d(no)}^k B_{d2(no)}^k + D_{d(no-no)}^k$$
$$f(x_n, u_n, u_{n+1(in)}) = C_{d(no)}^k \left\{ A_d^k x_n + B_{d1}^k u_n + B_{d2(in)}^k u_{n+1(in)} \right\} + D_{d(no-in)}^k u_{n+1(in)}.$$

then be described by local state-space systems, including switches permutations. For example, imposing a node of a priori unknown voltage it creates a virtual separation in the state-space equations. By considering Figure 1, the blocks of the electrical components and their connections correspond to a possible choice of nodal separations. The challenge is to find a way to compute this nodal voltage and complete the iteration of the virtually separated group equations.

In order to show how such technique works let us consider a system that is separated at some points with nodal connection points of, a priori, unknown voltage values. After discretization, by trapezoidal or higher order methods, each of these groups has a state-space equation in the following form:

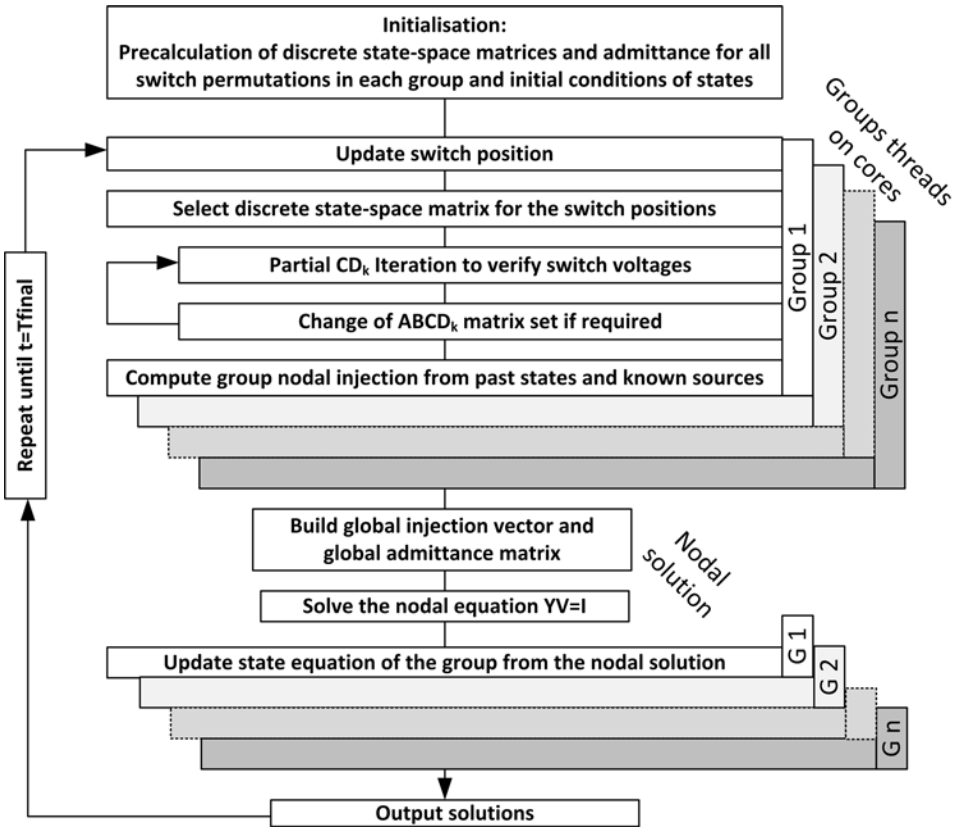
$$\begin{cases} x_{n+1} = A_d^k x_n + B_{d1}^k u_n + B_{d2}^k u_{n+1} \\ y_{n+1} = C_d^k x_{n+1} + D_d^k u_{n+1} \end{cases} \quad (30)$$

which can be further decomposed in the following way (see Box 5).where the subscript d indicates that we are dealing with discrete matrices, *in* refers to internal input/output of the group and *no* refers to nodal input/output of the group. These nodal quantities are unknown values to the group equations and have to be solved simultaneously for all groups using Thévenin equivalents and a nodal method. The nodal equations are built from the Norton equivalents and are found directly using the following relationships (see Box 6).

The interpretation of these equations can be made when $u_{n+1(no)}$ is chosen to be voltages and $y_{n+1(no)}$ currents. Then W^k has an admittance form which permits the construction of the system of nodal equations. A similar interpretation can be made if $u_{n+1(no)}$ is chosen to be currents and $y_{n+1(no)}$ voltages, (W^k has an impedance form then). $f(x_n,u_n,u_{n+1(in)})$ is a function of past time step states and inputs as well known sources at the current time step.

So as mentioned above, another fundamental issue to be considered for state-space nodal solvers is the impulse event detection. A key aspect of forced-commutated converter is that they are built

Box 7. SSN solver algorithm



in such way that no current discontinuity occurs in the devices. Typically, the switches will have anti-parallel diode connected to them, see Figure 4, or the configuration will be made with the necessary free-wheeling diodes to avoid over-voltage on the switching elements that would happen if an inductive current was stopped. At the simulation level, this creates so-called impulse events: when a forced switch opens or closes, the circuit topology sometimes induced an instantaneous closing or opening of another switch in the circuit. At the simulation level, this poses a challenge, especially in real-time where solver iterations are preferably avoided. When the system is described by time-segment linear state-space equations, the states have the useful properties that they cannot change instantaneously, i.e., to jump, when a switching action occurs. This property is also present in the groups of SSN, when the groups are made in such a way that the causal links of the impulse events are grouped together. This enables the SSN algorithm to check for this impulse conditions on a group by group basis in a very efficient way.

The complete SSN algorithm is synthetized in Box 7.

The switches status are updated at the beginning of the time step from known states and inputs of each subsystems. These inputs include the gate signals of power electronic systems. The impulse event detection is then made by checking that the new choice of matrices does not cause other switches to change status. As the switches are modeled in a binary manner and purely resistive, the switch status can be determined from gates and voltages (current is deducted from voltage then).

REAL-TIME HIL TESTS

In this section we present some real-time HIL simulation tests obtained on realistic railway applications. The fidelity of most of the models implemented in the HIL set-up has been validated through a comparison of non real-time simulation results with experimental data (Busco, et al., 2003). The main objectives of the HIL tests are the verification of the TCU both in normal operations and in fault conditions. The tests performed in normal conditions are important to obtain a preliminary confirmation about the capability of the train to satisfy the requirements imposed by the commitments. On the other side, the HIL tests performed under fault conditions, are essential to evaluate the capacity of the mechanical and electrical components of the equipment under control to comply with a particular stress event.

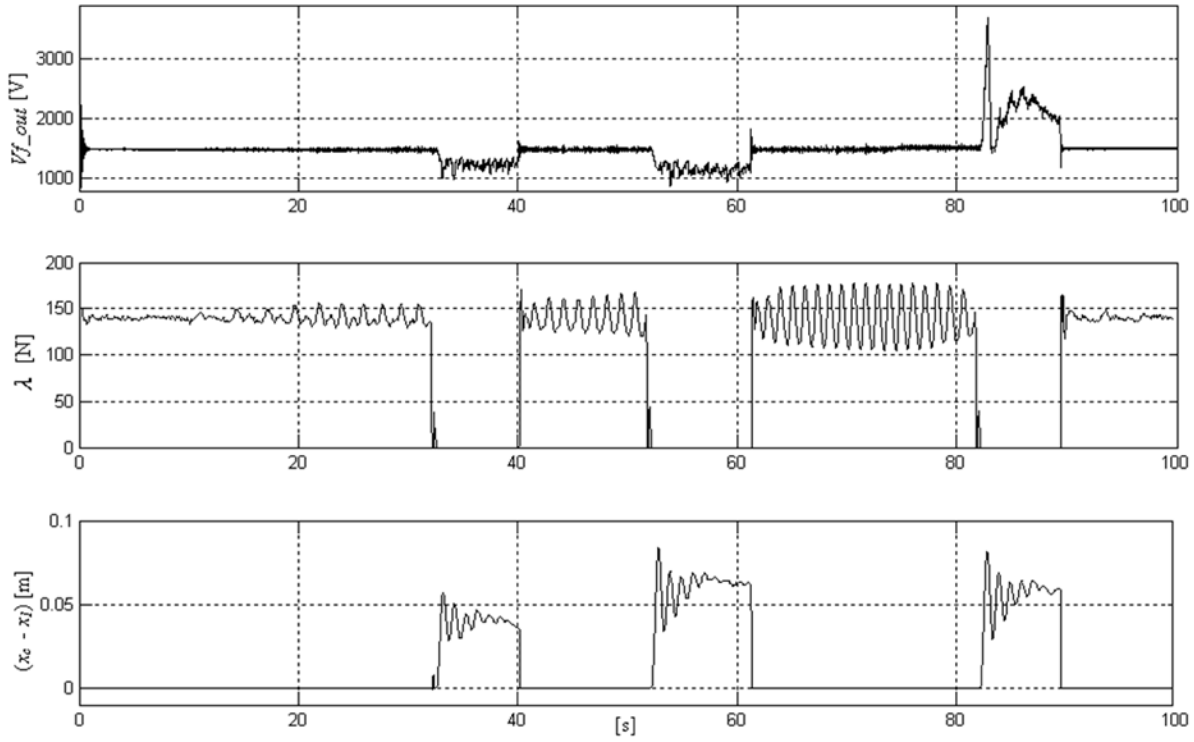
The following scenarios are considered:

- A missing contact of the pantograph;
- A stator winding disconnection of the induction motor;
- A driving/braking test;
- The opening of a contactor and, consequently, the activation of the protection system.

The first three tests are made by considering the traction system parameters corresponding to the Madrid Metro 7000/9000 series of AnsaldoBreda (http://www.ansaldobreda.it/). The metropolitan train consists of six cars, with total capacity of about 1260 passengers with maximum speed 120 km/h. The basic structure is an articulated frame which distributes load to the wheels depending on operating conditions, as well as offering stability, safety and comfort even when there are track or wheel wear problems. Each motor bogie has two transversally mounted motors. The four traction motors in each motor car are fed from a IGBT (insulated gate bipolar transistor) based inverter with a corresponding IGBT braking chopper. High switching frequency can be provided with local failure protection, featuring electronic circuitry that switches off the inverter before over-currents reach dangerous levels. The switching frequency of the IGBTs is sufficiently high to generate a low harmonic content waveform which drastically reduces low speed torque pulsing. The overhead line has a power supply continuous voltage which can range from 600 V to 1500 V. The power each motor is 198 kW, the frequency is 62.5 Hz.

The first test simulates temporary missing contacts between the pantograph and the catenary. Such situation, which can be also considered as a representation of the so called icing effect, is a typical critical operating condition to which the traction control system must efficiently react. Figure 7 shows the variations of the voltage at the output of the locomotive filter, the contact force and the detachment length. The oscillations occurring during the simulation reproduce the time varying catenary parameters, see (11). After 30 seconds three missing contacts of ten seconds are simulated: the no-zero detachment length demonstrates the reproducibility of this situation in the real time test. The disconnection time of 10 seconds allows to evaluate all the effects of the oscillations on the simulated variable x_c-x_1 representing the distance between the pantograph and the catenary due to the temporary fault of the hydraulic actuator of the pantograph, see (1)-(3) and Figure 2. This situation is also critical from the electrical point of view because of the fast reduction of the line filter voltage during traction (the first two missing contacts) and its increasing during braking (the last missing contact). In the first plot of Figure 7, more precisely in the first two events reported, it is shown the effects on the DC link of an electric arc due to the breakdown of the air between the catenary and the point of contact of the pantograph. In the second event, the train has a greater velocity with respect to the previous one. The effects of the disconnection are more evident both in the maximum distance

Figure 7. DC voltage link during a missing contact of the pantograph-catenary system

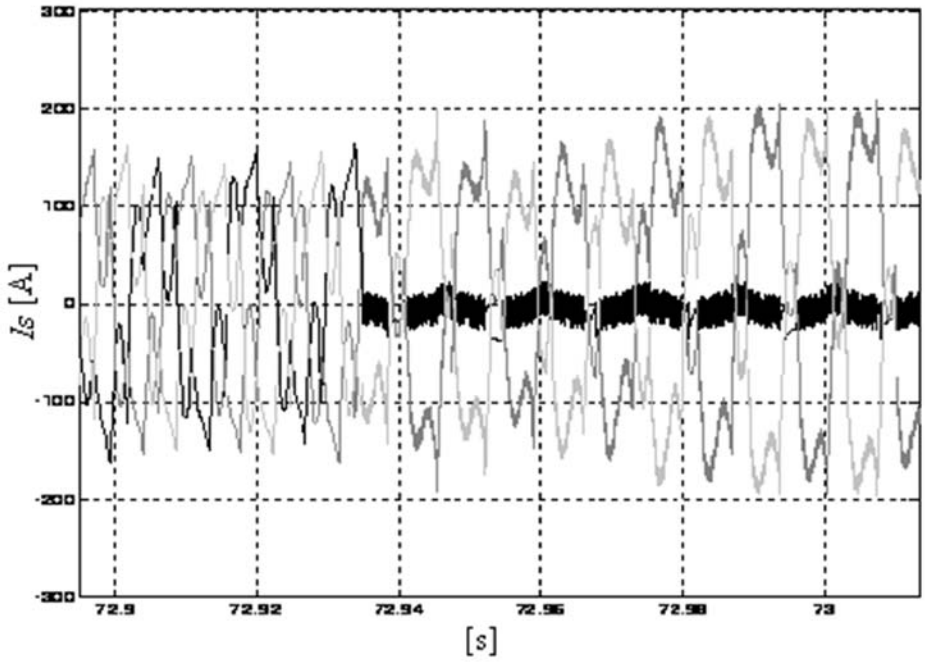


between the catenary and pantograph and the amplitude of the oscillation on the contact force between them. The velocity of train affects significantly the oscillation of adherence force but, anyway, this is sufficient to permit the conduction between pantograph and catenary. The disconnection is simulated by reducing, at different velocity and operative conditions, the pre-charge force F in the last equation of (1). In the third event the train is in breaking mode and the disconnection produces an increase of the DC link voltage due to the energy recovery of the motors until the activation of the braking chopper. This event shows the dual working mode: the first spike of DC voltage was limited by the rapid insertion of the dump load and then, after this transient, by the PWM modulation control of the chopper.

A very dangerous situation which eventually occurs in the operating conditions of an electrical locomotive consists in the disconnection of one motor phase. In this case the current in the faulty phase goes to zero causing an asymmetric behavior of the drive, see (22)-(29). In general, this fault is due to the inaccurate tighten of the cable. During the service, the mechanical vibrations could cause the disconnection of the phase cable with a consequent imbalance of the load. The currents flow through the connected phases producing, in the motor, two equal and opposite stator magnetic fluxes without generating any torque, Then, the electrical power absorbed from the inverter is converted to heat in stator winding causing an increase of their temperature. The simulation of this fault was permitted by the deactivation of the protection system in the TCU. This functionality monitors the unbalance of currents in the motor's phases and, in this case, commands the rapid shutdown of the inverter.

The real-time simulation is able to reproduce the effect of this fault on the electrical and mechani-

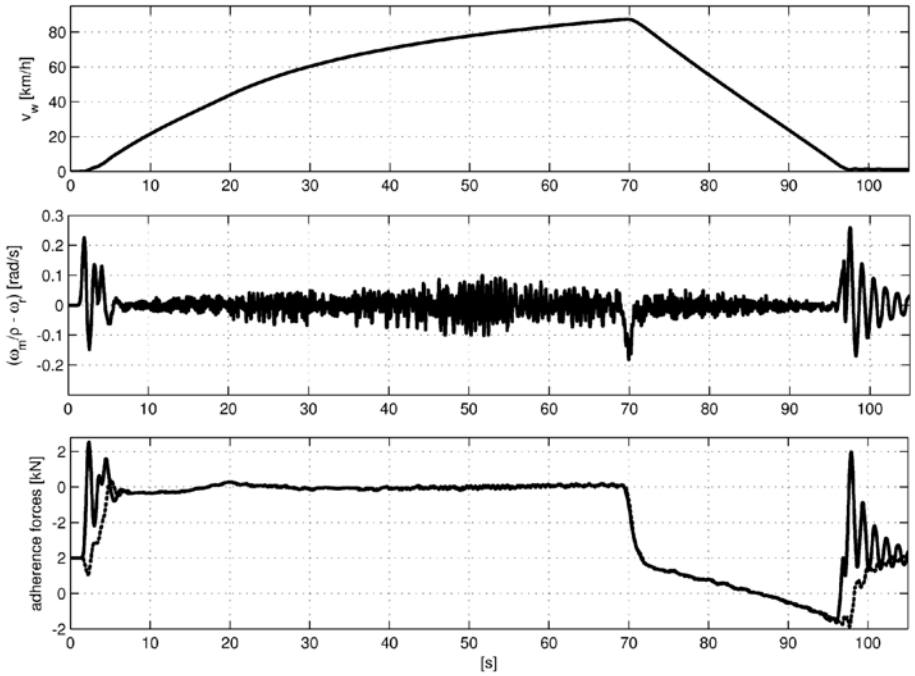
Figure 8. Stator currents during an open terminal fault



cal variables which are included in the real-time model presented in the previous section, see Figure 8. This example confirms the potentialities of the real-time HIL simulations. Indeed it is possible to simulate the TCU reactions to such unusual situation which is very difficult, expensive and dangerous to be tested on the actual train.

All control systems of modern electrical locomotives are equipped with the so called anti-slipping control strategy. Indeed the typical slipping phenomenon between wheels and rail, which allows the train motion in ordinary operating conditions, becomes undesired when the slipping speed increases over a certain bound which depends on environment and locomotive operating conditions. The real-time HIL simulation technique represents a valuable support to test the effectiveness of anti-slipping control strategies and to reduce the time needed for the controller calibration in the complex set-up used for testing the actual implementation. Figure 9 shows both the train speed profile (first plot) and the difference between left wheel and reducer speed (second plot), which represents the torsion of the shaft. In the last plot are represented the adherence forces of each wheel indicated with F_r and F_l, which are the load forces for the right and left wheel, respectively. The simulation results shows the complex behavior and the importance of slipping during fast traction and braking maneuvers, so as the effectiveness of the controller implemented in the real TCU which closes the loop with the real-time simulated locomotive. It is interesting to note that it is also possible to simulate variations of the profile of the adherence coefficient This scenario corresponds to the changing of the rail-wheel contact area, for example due to different weather conditions. The difference of the adherence forces (third plot in Figure 9) between right wheel (continuous line) and left wheel (dotted line) is due to the presence of the hallow shaft. In fact, the right wheel is connected to reducer with a more rigid and hollow shaft instead of the left

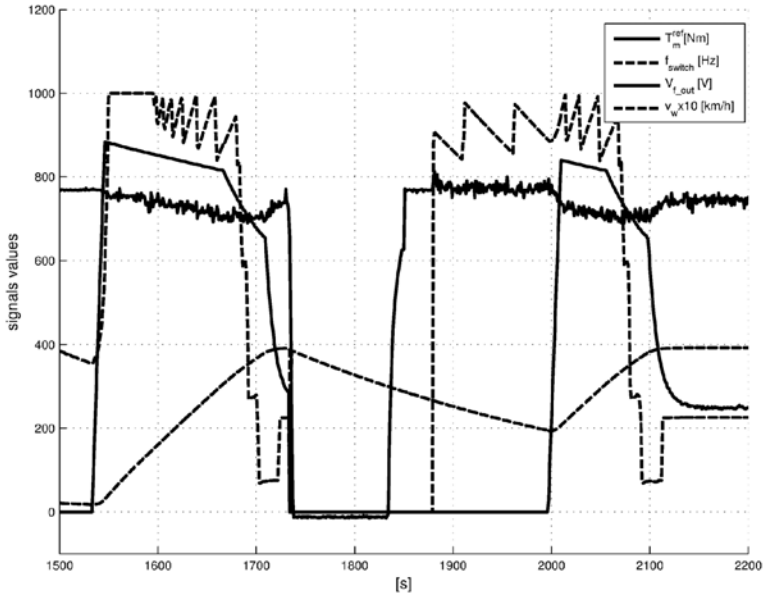
Figure 9. Traction/braking tests and slipping phenomenon



one which is connected by a solid shaft with the right wheel, see (21). The third plot in Figure 8 also shows the positive adherence force of in the traction phase and the negative adherence in the breaking phase.

The last test is made by considering the traction system parameters corresponding to the tram Sirio Atene of AnsaldoBreda. The tram is a bidirectional light rail vehicle, with maximum speed 70 km/h, composed by 5 bodies on 3 bogies, where two are motor bogies and one is trailer. The traction system is equipped, for each motor bogie, with an IGBT-inverter controlling both motors. The overhead line has a nominal continuous voltage of 750 V. By considering this case study it has been simulated the activation of the protection system and consequently the disconnection of the overhead line. Figure 10 shows the evolution of some variables of the real-time models. In particular, after the first acceleration time interval of around 170 seconds we simulated a motor over current: the protection system actuate a rapid disconnection of the overhead line, the TCU turns off the inverter and the filter capacitor discharges. When the protection system allows the restart process, the pre-charging of the filter capacitor is carried out and then, when the filter voltage is around its nominal values the inverter is turned on in coasting mode (zero torque applied) so as shown by the increasing of the switching frequency passing through different pulse modes. At 200 seconds from the start of the simulation the TCU goes in traction mode and the tram starts accelerating until the velocity limit (40 km/hour) is reached. Then the TCU goes in cruising mode. In the last part of the simulation a typical downhill situation is considered: the torque decreases while the tram speed increases. The proposed example demonstrates the potentialities of the real-time HIL system for the validation and verification of the on board software used for protection and safety of the traction system.

Figure 10. HIL simulation results for the tram Sirio Atene during a particular scenario



CONCLUSION

Real-time hardware in the loop (HIL) techniques represent an important tool supporting the verification and validation processes of the control software. The HIL technique is widely used since many years for aerospace and automotive transportation vehicles in order to test the control software in ordinary and unusual operating conditions, so as to study the behavior of special vehicles topologies. Only recently the application of this technique is showing its potentialities also for railway vehicles. In this chapter a railway case study was presented.

The core of a HIL system is represented by the model of the system to be simulated in real-time. In the first section the mathematical models of the most important parts of a typical electromechanical train propulsion system were presented and discussed: the pantograph-catenary system, the pre-charge circuit and the line filter, the braking chopper, the induction motor, the mechanical transmission system and the wheel-rail contact. The description of the different models and the design of their mathematical couplings allowed to present the modular approach required for the construction of the model of the system under control, which is, for our HIL application, the part to be simulated in real-time and to be interfaced with an actual traction control unit (the equipment under test). A similar modeling approach can be used for simulating different powertrain architectures or to represent with more details some specific phenomena. For instance, it was shown how slight modifications of some corresponding basic models allow to simulate the missing contact of the pantograph, the disconnection of a motor phase and the wheel-rail slipping phenomenon.

From the real-time simulation perspective, the most challenging part of the electromechanical train propulsion system is represented by the power electronic converters. In the second section the typical problems and solutions related to the real-time numerical integration of naturally stiff models of power electronic systems were presented. It has been shown that the use of the modular modeling

approach and the correspondence of sub-models characterized by different time-scales to different hardware processing units, such as field programmable arrays and central processing units boards, represents a key aspect for obtaining efficient real-time simulations.

Finally, in the last section some HIL verification tests for a train traction control unit were reported. The numerical results demonstrate that the proposed approach can be effectively used to analyze the behavior of electrical railway traction control unit and to evaluate the performances of the system, also under faults and in unusual or dangerous operating conditions.

REFERENCES

Abul Masrur, M., Chen, Z., Murphey, & Y. (2010). Intelligent diagnosis of open and short circuit faults in electric drive inverter for real-time applications. *IET Power Electronics, 3*(2), 279-291.

Allegre, A. J., Verhille, J. N., Delarue, P., Chattot, E., & El-Fassi, S. (2010). Reduced-scale-power hardware in the loop simulation of an innovative subway. *IEEE Transactions on Industrial Electronics*, *57*(4), 1175–1185. doi:10.1109/TIE.2009.2029519

Allotta, B., Pisano, A., Pugi, L., & Usai, E. (2005, December), *VSC of a servo-actuated ATR90-type pantograph*. Paper presented at the 44th IEEE Conference on Decision Control, Seville, Spain.

Bonta, D., Festila, R., & Tulbure, V. (2006, May). *The problem of speed measurements in the slip-slide control for electric railway tractions*. Paper presented at the IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania.

Bordas, C., Dufour, C., & Rudloff, O. (2009). *A 3-level neutral-clamped inverter model with natural switching mode support for the real-time simulation of variable speed drives*. Paper presented at the 8th International Symposium on Advanced Electromechanical Motion Systems, Lille, France.

Bose, B. K. (Ed.). (2006). *Power Electronics and Motor Drives-Advances and Trends*. Burlington, MA: Academic Press.

Busco, B., Marino, P., Porzio, M., Schiavo, R., & Vasca, F. (2003). Digital control and simulation for power electronic apparatus in dual voltage railway locomotive. *IEEE Transactions on Power Electronics*, *18*(5), 1146–1157. doi:10.1109/TPEL.2003.816198

Collina, A., Facchinetti, A., Fossati, F., & Resta, F. (2004). Hardware in the loop test-rig for identification and control application on high speed pantographs. *Shock and Vibration*, *11*(3-4), 445–456.

Di Pietro, C., Vasca, F., Iannelli, L., & Oliviero, F. (2010, October), *Decentralized synchornization of parallel inverters for train auxiliaries*. Paper presented at the International Conference on Electrical Systems for Aircraft, Railway and Ship Propulsion, Bologna, Italy.

Di Tommaso, P., Flammini, F., Lazzaro, A., Pellecchia, R., & Sanseviero, A. (2005, October). *The simulation of anomalies in the functional testing of the ERTMS/ETCS trackside system*. Paper presented at 9th IEEE International Symposium on High-Assurance Systems Engineering, Heidelberg, Germany.

Dommel, H. W. (1969). Digital computer solution of electromagnetic transients in single- and multiphase networks. *IEEE Transactions on Power Apparatus and Systems*, *88*(4), 388–399. doi:10.1109/TPAS.1969.292459

Dufour, C., Bélanger, J., & Abourida, S. (2003). Accurate simulation of a 6-pulse inverter with real time event compensation in ARTEMIS. *Mathematics and Computers in Simulation*, *63*(3-5), 161–172. doi:10.1016/S0378-4754(03)00072-7

Dufour, C., Dumur, G., Paquin, J. N., & Belanger, J. (2008, June). *A PC-based hardware in the loop simulation for the integration testing of modern train and ship propulsion systems*. Paper presented at the 39th IEEE Power Electronics Specialists Conference, Island of Rhodes, Greece.

Dufour, C., Mahseredjian, J., & Bélanger, J. (2011). A combined state-space nodal method for the simulation of power system transients. *IEEE Transactions on Power Delivery*, *26*(2), 928–935. doi:10.1109/TPWRD.2010.2090364

Facchinetti, A., & Mauri, M. (2009). Hardware in the loop overhead line emulator for active pantograph testing. *IEEE Transactions on Industrial Electronics*, *56*(10), 4071–4078. doi:10.1109/TIE.2009.2023632

(2004). Frontmatter. InNyce, D. S. (Ed.), *Linear Position Sensors: Theory and Application*. Hoboken, NJ: John Wiley & Sons.

Goodall, R. M. (2011, June). *Control for railways – active suspensions and other opportunities*. Paper presented at the 19th Mediterranean Conference on Control and Automation, Corfu, Greece.

Harakawa, M., Dufour, C., Nishimura, S., & Nagano, T. (2009, September). *Real-time simulation of a PMSM drive in faulty modes with validation against an actual drive system*. Paper presented at the 13th European Conference on Power Electronics and Applications, Barcelona, Spain.

Kalker, J. J. (1991). Wheel-rail rolling contact theory. *Wear*, *144*(1-2), 243–261. doi:10.1016/0043-1648(91)90018-P

Kassakian, J. G., Schlecht, M. F., & Verghse, G. C. (1991). *Principles of power electronics*. Reading, MA: Addison-Wesley.

Liu, W., Luo, G., Zhao, N., & Dou, M. (2009, September). *Design and HIL simulation of proportional compression salient-pole permanent magnet synchronous motor for electrical traction vehicle*. Paper presented at the 5th IEEE Vehicle Power and Propulsion Conference, Dearborn, Michigan.

Malvezzi, M., Allotta, B., & Pugi, L. (2008). Feasibility of degraded adhesion tests in a locomotive roller rig. *Institute of Mechanical Engineers*. *Part F: Journal of Rail and Rapid Transit*, *222*(1), 27–43. doi:10.1243/09544097JRRT108

Matar, M., & Iravani, R. (2010). FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients. *IEEE Transactions on Power Delivery*, *25*(2), 852–860. doi:10.1109/TPWRD.2009.2033603

Midya, S., Bormann, D., Schütte, T., & Thottappillil, R. (2009). Pantograph arcing in electrified railways - mechanism and influence of various parameters - Part I: with DC traction power supply. *IEEE Transactions on Power Delivery*, *24*(4), 1931–1939. doi:10.1109/TPWRD.2009.2021035

Mohan, N., Undeland, T. M., & Robbins, W. P. (Eds.). (2003). *Power Electronics: Converters, Applications, and Design*. John Wiley and Sons.

Pedicini, C., Vasca, F., Iannelli, L., & Jonsson, U. (2011, December). *An overview on averaging for pulse-modulated switched systems*. Paper accepted for presentation at the 50th IEEE Conference on Decision and Control, Orlando, Florida, USA.

Pugi, L., Malvezzi, M., Tarasconi, A., Palazzolo, A., Cocci, G., & Violani, M. (2005b, August). *HIL simulation of WSP systems on MI-6 test rig*. Paper presented at 19th Symposium of the International Association for Vehicle System Dynamics, Milan, Italy.

Pugi, L., Rinchi, M., Malvezzi, M., & Cocci, G. (2005a, July). *A multipurpose platform for HIL testing of safe relevant railway subsystem*. Paper presented at the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA.

Ren, W., Sloderbeck, M., Steurer, M., Dinavahi, V., Noda, T., & Filizadeh, S. (2011). Interfacing issues in real-time digital simulators. *IEEE Transactions on Power Delivery*, *26*(2), 1221–1229. doi:10.1109/TPWRD.2010.2072792

Terwiesch, P., Keller, T., & Scheiben, E. (1999). Rail vehicle control system integration testing using digital hardware-in-the-loop simulation. *IEEE Transactions on Control Systems Technology*, *7*(3), 352–362. doi:10.1109/87.761055

Vasca, F., Camlibel, M. K., Iannelli, L., & Frasca, R. (2009). A new perspective for modeling power electronics converters: complementarity framework. *IEEE Transactions on Power Electronics*, *24*(2),456–468. doi:10.1109/TPEL.2008.2007420

Chapter 11
Hardware-In-the-Loop Testing of On-Board Subsystems:
Some Case Studies and Applications

Luca Pugi
University of Florence, Italy

Benedetto Allotta
University of Florence, Italy

ABSTRACT

Hardware In the Loop testing is a very powerful tool for the development, tuning, and synthesized homologation of safety-relevant on-board subsystems and components.

In this chapter some case-studies, based on typical topics of industrial research for railways, are introduced in order to emphasize some aspects of the mechatronic design with a particular attention to the integration of actuation systems into rig design.

SYSTEM IN THE LOOP TESTING: INTRODUCTION

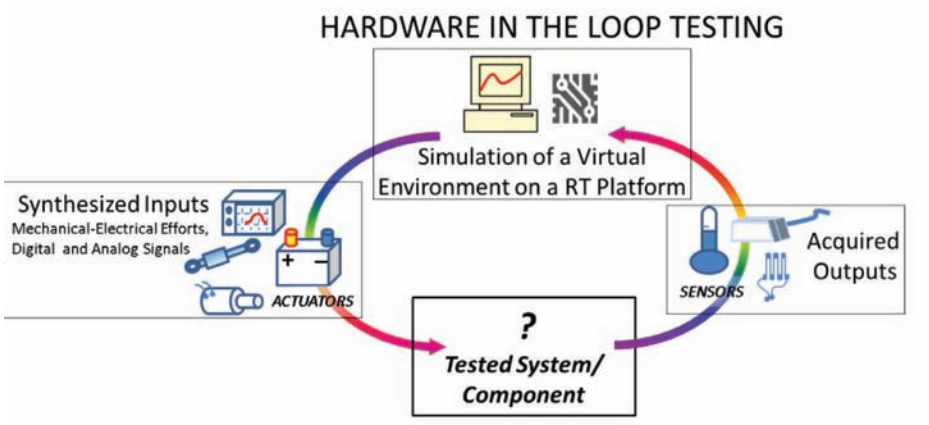
HIL (acronym of Hardware-In-the-Loop) testing is a technique commonly used for the development of advanced mechatronic systems in order to test them by simulating in a realistic way strong cross-coupling effects between the tested subsystem/component and the surrounding environment. The original term Hardware- In-the-Loop can be extended to a more general "System-In-the-Loop," where "System" might include electro-mechanical and electronic hardware, and/or software, and/or firmware, or all of them.

As visible in the simplified scheme of Figure 1, the interaction of the tested component with the surrounding environment is reproduced by a closed loop system composed of sensors, actuators and a real-time model able to calculate and reproduce a virtual testing environment which approximates real operating conditions: sensors are used to evaluate the response of the tested component; measurements from sensors are used by the real-

DOI: 10.4018/978-1-4666-1643-1.ch011

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1.HIL testing, simplified scheme



time model to evaluate the corresponding evolution of the simulated environment. Actuators and signal generators are then used to provide inputs to the tested components in order to reproduce the calculated dynamical behavior of the simulated environment, thus closing the test-loop. Since the simulation loop is composed of real components, a finite delay between sensor measurements and synthesized inputs is involved; for HIL testing this delay has to be negligible respect to the typical time constants of the simulated environment. Also precision and bandwidth of sensors and signal generators are very important in order to obtain near to realistic testing conditions.

The application of HIL testing to safety-relevant railway on-board subsystems is highly recommendable for many reasons:

- **Different Operating Scenarios:** The testing conditions are reproduced by a numerical model so it is quite simple to change the testing conditions in order to reproduce different operating scenarios, even the least probable (but possible!) and potentially really dangerous in the real application.
- **Costs and Time Consumption:** Experimental tests of on-board subsystems involve the availability of testing trains, testing circuit/line and qualified human resources. As a consequence, this kind of experimental activities are quite expensive and time consuming. The availability of specialized test rigs for repeatable testing campaigns involves lower costs and lower time consumption.
- **Availability and Logistical Limitations:** Testing on railway lines involve the availability of resources which are often precluded to developers especially in the design and tuning phase of innovative systems. As an example, preliminary testing and tuning of an innovative traction control system may involve the availability of a vehicle or a complementary subsystem which is still in the construction phase.
- **Safe Testing Conditions:** With HIL testing, it is possible to simulate potentially dangerous conditions which can be quite difficult to be reproduced on line test for safety reasons: as an example, extreme testing conditions, which involve the risk of severe accidents.

On the other hand, validation of a proposed HIL testing procedure is a quite complex matter since it involves the qualification of a complete test rig composed of data acquisition systems, numerical models running in real-time and actuation/signal

generation systems. HIL testing devices should be verified/validated by comparing the obtained results with more sophisticated codes developed for off-line simulation and with experimental data.

In addition, given the closed-loop nature of the rig plus component-to-be-tested system, it is mandatory to address and solve the classical trade-off problem between performance (in this case "performance" is to be intended as "high fidelity") and robust stability of the overall system (Malvezzi,2008).

The availability of a network of recognized-homologated test rigs with commonly accepted procedures based on regulations is fundamental to simplify the validation /homologation process of an HIL test rig for a specific application.

In this chapter, case studies-applications are proposed to introduce some general matters related to HIL simulation of railway components and subsystems. In particular three case studies of increasing complexity are chosen as examples:

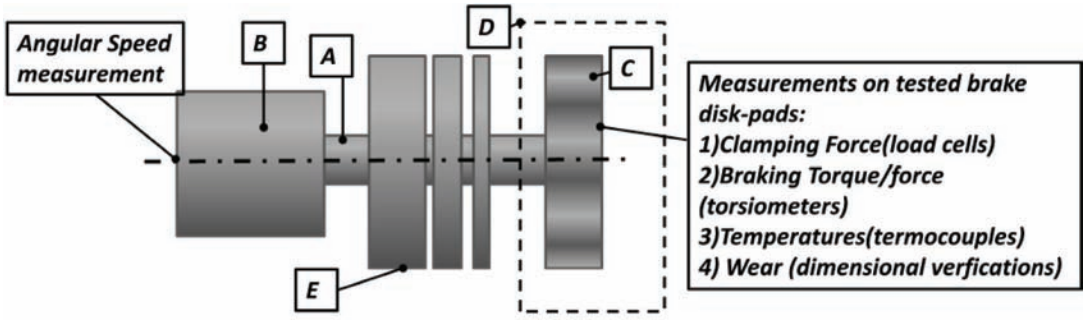
- **Simulation of Vehicle Inertia for yhe Testing of Brake Pads and Discs According to Enforced Regulations (UIC541-3, 2010),(UIC541-4,2007):** This simple application is used to introduce some features of HIL testing and the concept of mechanical impedance simulation.
- **Hil Simulation of Railway Pantographs:** Dynamical test rig for HIL testing of pantograph are diffused among many research institutions, (UIC541-3, 2010),(Facchinetti, 2004), (Mpanda, 2009),(Allotta, 2009). This is a good example to introduce a rig for the simulation of passive/active suspension systems and the troubles related to the limited bandwidth of sensors and actuators used for the simulation.
- **Mutual Interaction of Safety-Relevant Subsystems with Degraded Adhesion Conditions:** Many safety-relevant on-board subsystems like WSP systems, odometry algorithms of ATP/ATC systems and Traction Control-Antiskid interact with the rotational behavior of wheels and axles. In particular when traction-braking with degraded adhesion conditions occur, at least two or three different on-board mechatronic subsystems may be mutually affected. As example a pneumatic-electric braking with degraded adhesion conditions involves the modulation of the efforts due to brake cylinders and to motors; modulation of brake efforts is performed by wsp(wheel slide protection system) and/or by traction control systems; also train position estimation may be affected since the measurements of axle rotational speed is an input for some of the most common odometry boards/algorithms such as the Italian SCMT (Trenitalia, 2000). As a consequence, Trenitalia (Toni, 2003), (Pugi, 2006) and many other research and industrial subjects have developed rigs for the HIL simulation of this kind of conditions.

Considering the wide variety of technical troubles and scientific research topics involved, the presented case studies are used to introduce the multi-disciplinary approach which is needed in the design of HIL testing systems especially when strong coupling between different physical effects or systems are involved. In particular some topics related to the implementation of real time code including distribution and management of multiple tasks, balancing of performances vs. computational loads and consequently the choice of the hardware used for code implementation have been deliberately neglected considering the availability of a specific literature and the general purpose of the work. Also another interesting topic that for the limited space available is only partially introduced in this work is the detailed verification/debugging of the functionality of the controllers used on many on board subsystems, including software/firmware implementation respect to desired specifications.

Table 1. specifications of an existing test rig (Pugi,2002)

Mechanical features		Electrical features		Pneumatic and system features	
Test rig inertia (No flywheels)	625 kgm^2	Max power	391 kW	Pressure rise time (braking clamping force)	4 s (±0.2)
Flywheel 1 inertia	250 kgm^2	Nominal torque	6192 Nm	Average initial delay time (clamping force application)	0.1 s
Flywheel 2 inertia	650 kgm^2	Max speed	1500 rpm	Max braking torque	7800 Nm
Flywheel 3 inertia	1625 kgm^2	Drive working Frequency	10 kHz	Clamping force precision (relative)	±1.25%

Figure 2. Typical layout of a test rig for the testing of brake disks and pads (A=shaft B=motor, C=tested disc/pad,D=conditioned testing room, E=flywheels



AN INTRODUCTIVE EXAMPLE: TESTING OF RAILWAY DISKS AND BRAKE PADS

According to latest fiches of UIC (the International Union of Railways) (UIC541-3, 2010), (UIC541-4, 2007), disks brakes and pads have to be tested on a rig which is able to reproduce on disks mechanical and thermal loads due to vehicle inertia. Main elements of typical test dynamometer test rig are shown in Figure 2: the rig is composed of a shaft (A) moved by an electric actuator (B). The actuator is usually a separately excited DC machine or a Synchronous motor, actively controlled by a static converter drive. The brake(C), which has to be tested, is usually placed at the end of the shaft simulating the axle rotational motion. Clamping forces are generated by a pneumatic actuator with closed-loop force control in order to assure repeatability of testing conditions. A fan system simulates the real environmental conditions (temperature, humidity, air speed, etc.) in the test area (D). Vehicle inertial properties are simulated with the contribution of both a variable array of flywheels (E) and a compensating torque provided by the electric actuator (B), usually in a direct-drive configuration without any gearbox or othe mechanical transmission. Forces, temperatures and many other physical parameters of the disk are measured in real-time. In Table 1 some data referred to an existing test rig (Pugi, 2002) are shown.

Typically two main kinds of tests are performed:

- **Stop Braking Tests:** Contributions of external forces such as gravitational forces are not considered (no slope applied); the vehicle is simply modeled as a known inertia; a stop braking with assigned vehicle

inertia, starting speed and clamping forces on brakes is performed. Cycles of simulated braking maneuvers are repeated in order to verify performances in different operating conditions.

- **Constant Speed Tests:** In this case brake clamping forces are regulated in order to keep a constant speed while the rig impose a known mechanical power/thermal load dissipated on the disk. Typically this is a prolonged test in which the endurance of disks and pads to constants heavy thermal loads is verified.

In order to perform stop braking tests, loads due to vehicle inertia have to be reproduced.

The equivalent part of vehicle inertia I_i loading the *i-th* brake disk can be calculated according to (UIC541-05,2005). assuming good adhesion conditions (no-slipping-skidding) between wheel and rail interface.

τ_i is the kinetic ratio between disk rotational speed ω_r of the axle corresponding to the i-th brake disk and the longitudinal speed of the vehicle $\dot{x}$ as defined by (1).

$$\tau_i = \frac{\omega_r}{\dot{x}} = \frac{\partial \theta_r}{\partial x}$$
$$\text{where } \theta_r = \text{ angular rotation of the wheel}$$
$$x = \text{carbody/train longitudinal motion} \tag{1}$$

The kinematic energy that has to be dissipated by the i-th disk is defined by (2)

$$E_c = \frac{1}{2} I_i \omega_r^2 \tag{2}$$

The lagrangian equivalent torque M_j, corresponding to a generic external force F_j acting on vehicle carbody, can be expressed according (3):

$$M_j = F_j \frac{\partial x}{\partial \theta_r} = \frac{F_j}{n_{disks}\tau_i}$$
$$n_{disks} = \text{number of disks/vehicle}$$
$$\text{(equal distribution of braking efforts on every disk)} \tag{3}$$

Relation (3) it's calculated assuming that braking efforts are equally distributed on the brakes of the vehicle. This condition is not mandatory but it's commonly verified in many applications in order to obtain an homogenous wear rate of the friction components of the vehicle (disks, pads,etc.).

Appling the lagrangian formulation, the simulated behavior of the disk on the vehicle is described by (4):

$$\frac{d}{dt}\left(\frac{\partial E_c}{\partial \dot{\theta}_r}\right) - \left(\frac{\partial E_c}{\partial \theta_r}\right) = M_i + \sum_{j=1}^{n} M_j \Rightarrow I_i \dot{\omega}_r = M_i + \sum_{j=1}^{n} M_j$$

where: ω_r =rotational speed of the tested disk/pads on the i-th axle

Mi=braking torque exerted by the tested disk/pad (4)

The dynamical behavior of the test rig is described by (5) (see Box 1).

By merging Equations (4) and (5), the torque M_e exerted by the actuator to emulate the desired dynamical behavior can be calculated (6):

$$\frac{I_r}{I_i}\left(M_i + \sum_{j=1}^{n} M_j\right) = M_e + M_i + M_d \qquad \Rightarrow$$
$$M_e = M_i\left(\frac{I_r - I_i}{I_i}\right) + \frac{I_r}{I_i}\sum_{j=1}^{n} M_j - M_d \tag{6}$$

If M_i and M_d are measured/estimated on line it is possible for the rig to simulate/emulate the mechanical behavior of the disk on the simulated vehicle; the resulting control system described

Box 1.

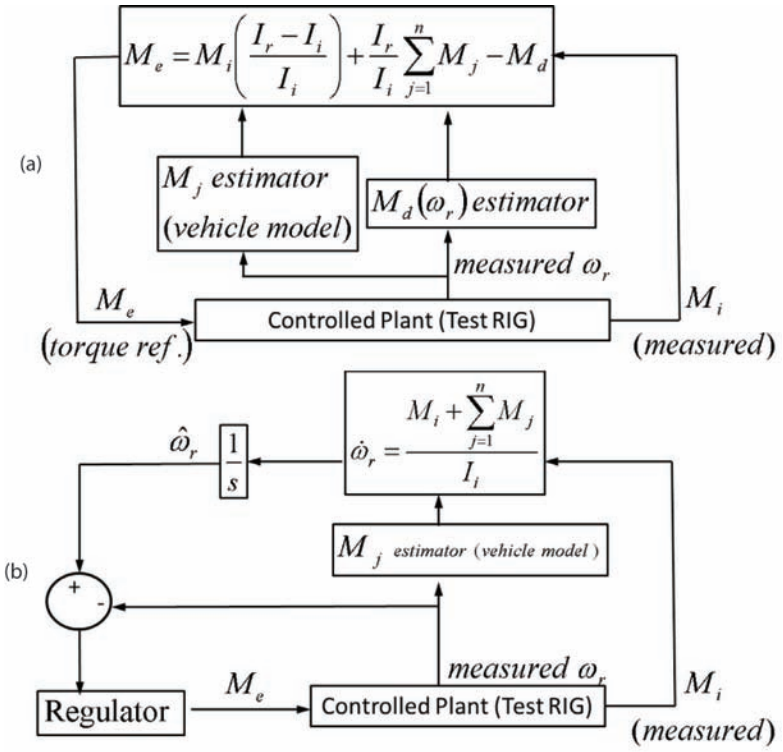
$$I_r\dot{\omega}_r = M_e + M_i - \overbrace{\left(k_2\left|\omega_r\right|\omega_r - k_1\omega_r - k_0 sign(\omega_r)\right)}^{\text{internal friction or ventilation losses}} = M_e + M_i + M_d$$

$where: I_r$ = total rotational inertia of the test rig

M_e = torque provided by the rig electrical motor

M_d = torque due to internal dissipation of the rig (5)

Figure 3. (a) Impedance control, control scheme (b) Impedance control with inner velocity loop



by the scheme of Figure 3 (a) can be seen as a particular application of the more general control strategy usually called impedance control or impedance simulation (Sciavicco,1996).

Unfortunately the application of this kind of control strategy in the form proposed in figure 3/a may lead to performance/robustness troubles which are related to uncertainties mainly concerning the following aspects:

1. The estimation of internal losses of the rig M_d is obviously approximated and affected by uncertainties.
2. Torque control of the actuator cannot be ideal, introducing errors between the desired value of M_e and the compensating torque really applied to the shaft.
3. Additional, un-modeled dynamics exist such as torsional/flexural behavior of mechanical components (shaft, joints, etc).

Box 2.

$$I_i\dot{\omega}_r = M_i + \sum_{j=1}^{n} M_j \qquad \Rightarrow \qquad \dot{\omega}_r = \frac{M_i + \sum_{j=1}^{n} M_j}{I_i} \qquad \Rightarrow \qquad \hat{\omega}_r = \int_0^t \dot{\omega}_r dt \tag{7}$$

In order to partially reject the effects of disturbances and in particular of friction, an impedance control approach with an inner speed loop is one of the most diffused strategies:

Directly from (4) is possible to calculate the reference speed profile $\hat{\omega}_r$ as shown in Equation (7), corresponding to a perfect emulation of the desired dynamics (see Box 2). $\hat{\omega}_r$ calculated according to (7) is then imposed as a reference profile for an inner loop; since the internal speed loop is quite robust against friction disturbances the estimation of M_d is often neglected, the resulting control scheme is visible in Figure 3 (b).

More accurate control strategies can be applied to the simulation of mechanical impedance such as sliding mode control (Slotine,1991), however to compare and to evaluate performances the most used indexes are relative and absolute error on simulated inertia $\Delta I_a, \Delta I_r$, defined according (8):

$$\begin{aligned} \Delta I_a &= I_i - \frac{M_i + \sum_{j=1}^{n} M_j}{\dot{\omega}_r} \\ \Delta I_r &= \frac{\Delta I_a}{I_i} \end{aligned} \tag{8}$$

The implementation of (8) may be impractical and affected by noise problems so a discrete implementation (9) with a minimum sampling frequency of 60Hz is often preferable and also recommended by existing standards (ERRI B126 RP18, 2000):

$$\begin{aligned} \Delta I_a(t_n) &= I_i(t_n) - \frac{\left(M_i + \sum_{j=1}^{n} M_j\right)\left(\theta_r(t_n) - \theta_r(t_{n-1})\right)}{\left(\omega_r^2(t_n) - \omega_r^2(t_{n-1})\right)} \\ \Delta I_r(t_n) &= \frac{\Delta I_a(t_n)}{I_i(t_n)} \\ where: \theta_r(t_n) &= \int_0^{t_n} \omega_r dt \\ t_n - t_{n-1} &\le \frac{1}{60 \cdot 2 \cdot \pi}\,[s] \end{aligned} \tag{9}$$

Further investigation regarding the choice of the optimal inertia I_r of the rig leads to more general considerations concerning the simulation of a mechanical impedance for Hardware in the loop testing:

1. Actuator forces, in this application M_e, are influenced by the difference between the simulated system and the real one. In particular for testing of friction material the difference between I_r and I_i influences the value of M_e calculated according to (6). In Figure 4 (a) the value of the required M_e assuming a test rig whose main features are described in Table 1 is shown. The torque is computed as a function of I_i and vehicle speed, considering a wheel with a radius of 0.65m and a constant vehicle deceleration of 1 ms^{-2} because M_e calculated according (6) is also a function of $\dot{\omega}_r$ which is a function of the assumed longitudinal deceleration $\ddot{x}$ and wheel radius r_w. In Figure 4 (b) same calculation of figure/a is performed considering a different wheel radius of 0.45m. As

clearly visible in figures, more demanding performances are associated to greater radius corresponding to heavy locomotives whose wheel diameters typically lays in the range of 1.2.1.3 meters

2. Since the simulated rig has three flywheels, rig inertia I_r is assumed to be modified in order to be closer to the value of the desired I_i. The possible flywheels discrete combinations are seven as shown in Figure 4 (a). Amplitude of control efforts has not only consequences on costs and encumbrances of actuators but also on bandwidth since dynamic performances of power systems usually decrease with their size.
3. Over-Bandwidth considerations: the actuator and, more generally, the control system used to simulate the impedance of a virtual (non existing) system has a finite Bandwidth. Usually the Bandwidth of the system is chosen according to specifications that reflect the matter that over a certain frequency the response of the tested component has not to be investigated or is not influenced by the inputs provided by the rig. However an expert user/designer has to be aware that the dynamical behavior of the rig over its maximum bandwidth may be very different from the corresponding virtual system reproduced by the testing device.

Acceptance Criteria for Brake Tests

Criteria for the acceptance of tested brake components depends from both international standards and additional specifications which are sometimes imposed by customer specifications.

Typically for both friction components of the brake (pads and disks) the most important parameters that have to be verified are the following:

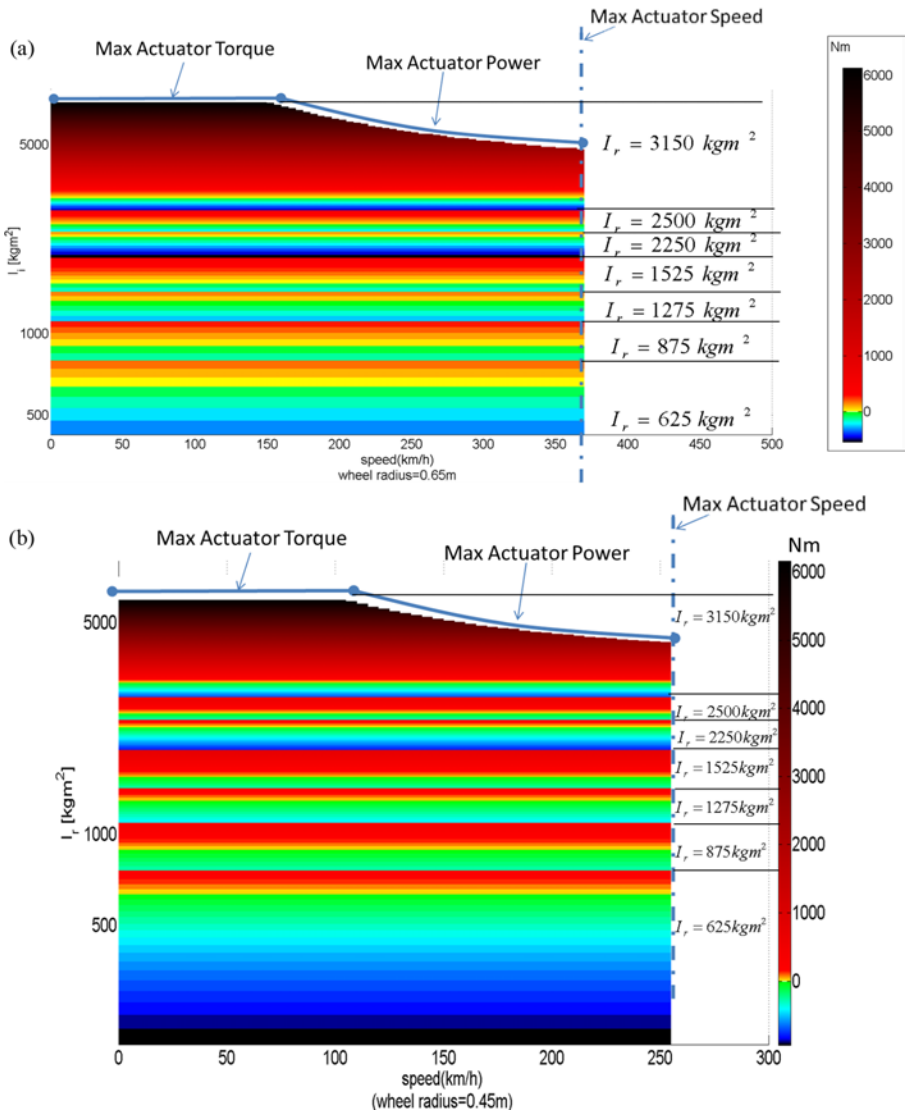
- **Friction Coefficient:** Braking performances depends on the stability of the value of the friction coefficient. For this reason tests are repeated considering different working conditions (speed, dissipated energy, braked inertia, cooling conditions etc.) in order to verify the stability of braking performances.
- **Wear Rate and Tribological Properties:** Friction surfaces are subjected to wear. Life of the component is usually determined by wear. As consequence, the measurement of wear in different working condition is an important parameter in order to evaluate disks or pads.
- **Mechanical Stability:** Pads and Disks are subjected to repeated heavy thermal and mechanical loads which may negatively affect structural and dimensional stability of the component that has to be verified.
- **Additional/Environmental Requisites:** Many complementary measurements observation may be performed such as noise measurements, chemical analysis,

HIL SIMULATION OF RAILWAY PANTOGRAPHS

On electrified railways current is usually provided by an overhead contact wire suspended to a catenary (the compound made by contact wire and catenary is briefly said “catenary”). Current is collected through a sliding contact assured by metal/carbon strips/shoes raised by an articulated suspension system called pantograph as visible in the simplified scheme of Figure 5 (a).

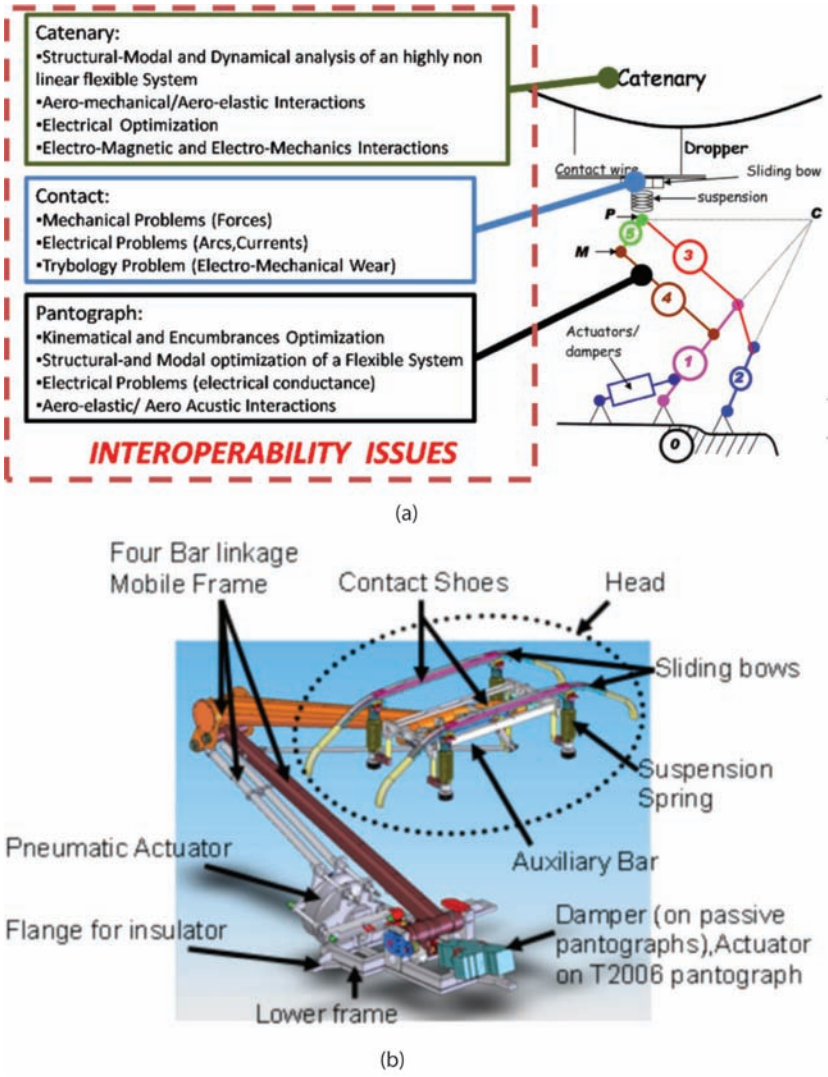
The pantograph, as visible in Figures 5 (a) and (b), usually features two contact shoes placed on sliding bows that are linked to the head of the mobile frame trough a suspension system with several degrees of freedom. The suspension system is design in order to reduce contact force fluctuations due to the pantograph-catenary interaction. Usually the moving frame is realized by a four bar linkage (symmetric pantograph are less diffused for high speed trains) optimized in

Figure 4. (a) Calculation of M_e as a function of vehicle speed and I_i (vehicle wheel radius=0.65m supposed vehicle deceleration of 1 ms-2) (b) Calculation of M_e as a function of vehicle speed and I_i (vehicle wheel radius=0.45m supposed vehicle deceleration of 1 ms-2)



order to obtain a vertical trajectory of the head for a total run of about 2-2.5 meters. In order to achieve better contact quality, horizontal alignment of the head is usually assured by an auxiliary bar. In Figure 5 (a) the kinematical scheme of a generic pantograph is showed: the auxiliary bar (member 4) creates an additional four bar linkage with members 5, 3 and 1. Kinematical approximations introduced by this second four bar linkage may be excessive. As a consequence, the revolute joint between the head and the auxiliary bar (M in Figure 5 (a)) may be substituted by a cam constraint (Ansaldo ATR95) or, more often, by inserting damped-elastic compliances as bushings or other elastic elements. In order to lift up the pantograph and to assure a known

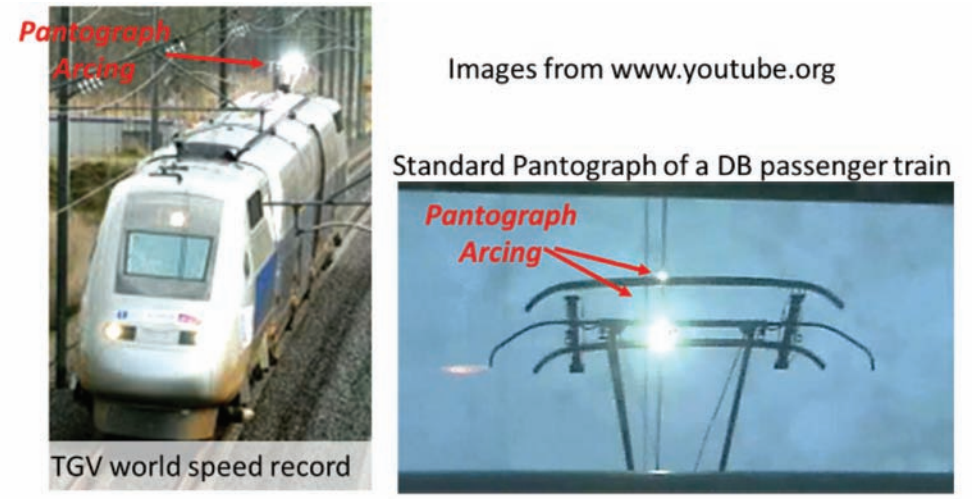
Figure 5. (a) Catenary-Pantograph interaction main elements and kinematical features (b) Pantograph main elements and nomenclature(Allotta,2009)



static force, the pantograph has a pneumatic actuator (working typically at 3.5-4.5 bar). In order to obtain a constant transmission ratio from air pressure to static force between sliding surfaces, the geometry of kinematic coupling between actuator and pantograph is optimized. One or more dampers are usually placed between the mobile frame and the lower frame that is constrained by electrical insulators to the train roof.

Quality and reliability of pantograph-catenary interaction and, consequently, of current collection is a crucial factor affecting performance and availability of traction systems and even of the whole line considering the possibility of critical failure of the overhead line. Problems arise also from arcing between the sliding surfaces of contact strips and catenary wire causing accelerated wear and electromagnetic noise that may disturb many safety relevant subsystems such as signaling, traffic management and communications. In Figure 6 some photos showing the typical "flash effect" associated to contact losses are shown.

Figure 6. Arcing, Some images taken from videos available on the web (www.youtube.org,2010)



Since various electrification standards are diffused in Europe and more generally world-wide, there is also the exigency of managing the interoperability of trains traveling under lines with different electrical and mechanical features.

As a consequence, pantograph-catenary interaction is a topic of industrial research and the object of international standards such as (Directive 2008/57/EC, 2009),(EN50206-1, 2009),(EN50367, 2010) as an example.

Hardware in the loop testing may be very useful to optimize pantograph-catenary interaction (modifications of existing solutions) and to develop innovative pantographs reducing time and resources consumption as described in (Baldauf, 2001), (Facchinetti, 2004), (Mpanda, 2009), (Allotta, 2009), (Allotta, 2008).

As visible in the scheme of Figure 7 (a), the simulation of the interaction between pantograph and catenary involves modeling and simulation of various multi-physical phenomena associated to different space and time scales. So it is quite difficult to reproduce all these aspects/phenomena in a single test rig. In particular three different aspects are studied on corresponding test rigs:

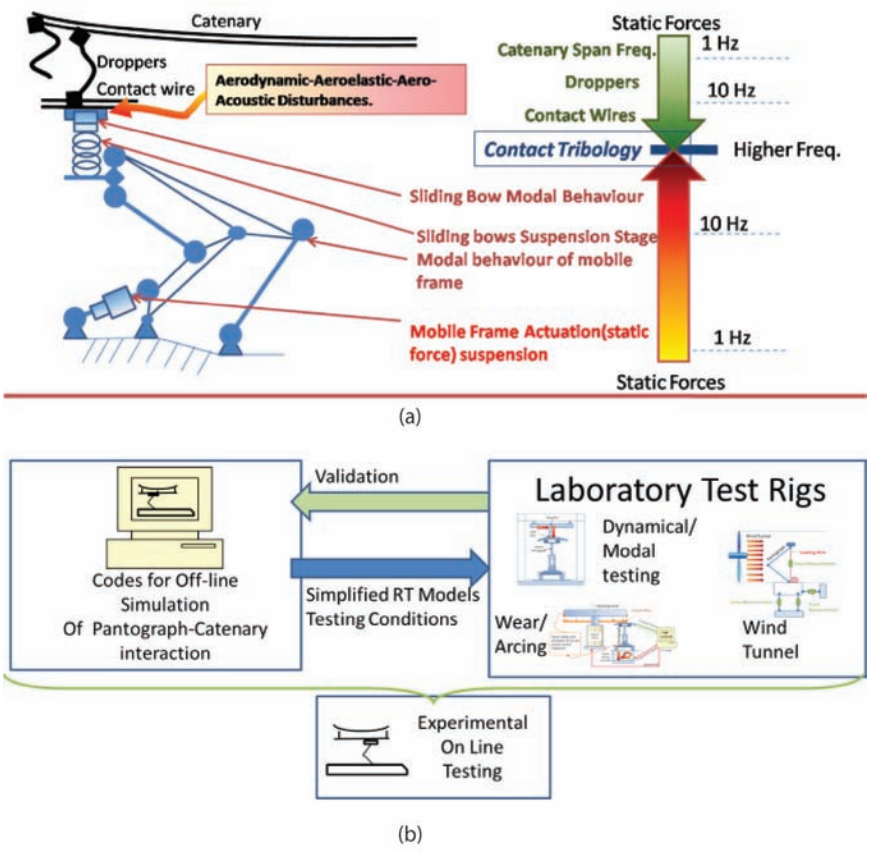
1. Dynamical and modal testing of Pantographs
2. Wear of sliding Surfaces and arcing
3. Wind Tunnel for aerodynamic, aero-elastic and aero-acoustic optimization

The three different kinds of tests are complementary and useful for the development of complete codes for off-line simulation of the whole pantograph-catenary interaction. Since Hardware-In-the-Loop simulation involves the availability of simplified models to support real-time simulations of the virtual operating conditions, there is a mutual benefit in the combined use of HIL simulation and offline models of pantograph-catenary interaction: experimental results from rigs may be used to refine offline models from which faster and simpler models for hardware in the loop testing can be obtained as visible in the simplified flow-chart of Figure 7 (b).

Dynamical and Modal Testing of the Pantograph

Dynamical response of actuators and suspensions of the pantograph is tested by imposing force/

Figure 7. (a) Different physical phenomena and time scales involved in pantograph catenary interaction (b) Synergy between laboratory test rigs and codes for Off-line Simulation



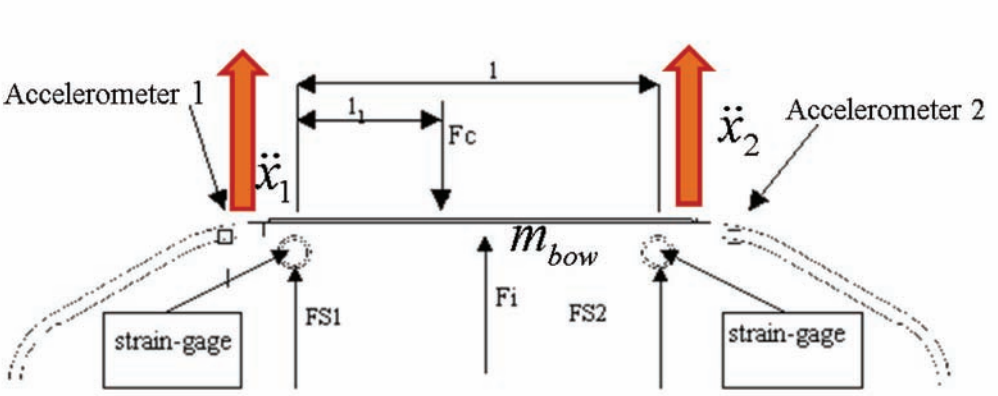
displacement patterns to the pantograph head as proposed, for example, by the testing procedures adopted in Italy by RFI (RFI, 2008). The dynamical response of the pantograph is evaluated using statistical operators of the contact force profiles $F_c(t)$ measured on pantograph head. Typical $F_c(t)$ measurements are performed using load cells and accelerometers placed on the pantograph sliding bows in order to compensate inertial effects. A typical sensor layout for a pantograph sliding bow is shown in Figure 8. Also the kinematical behavior of the system as for example suspensions and pantograph head run are usually measured.

Considering the typical sensor layout proposed in Figure 8, normal contact force value F_c and contact position along the sliding bow l_i can be evaluated according to (10) assuming the sliding bow as a rigid body:

$$\begin{aligned} & F_i + F_{s1} + F_{s2} = F_c \\ & F_c l_1 = F_{s2} l + F_i l_2 \\ & where\ F_i = -m_{bow}\left(\ddot{x}_1 + \ddot{x}_2\right) \end{aligned} \tag{10}$$

This kind of analysis is usually limited to a bandwidth of 20 Hz which is a commonly accepted dynamic limit for this kind of measurements. Higher bandwidth may be achieved using distributed sensors systems and estimation algorithms able to take into account the deformation of the sliding bows.

Figure 8. Contact force measurements and typical sensing layout of a pantograph sliding bow

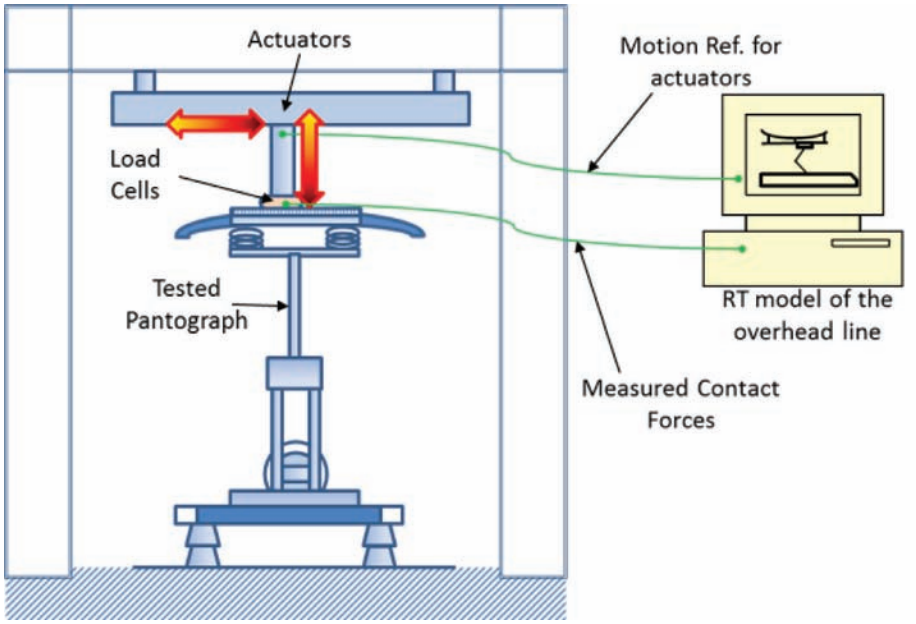


For HIL testing the imposed motion/force patterns used in conventional test rigs are substituted by a simulation loop shown in Figure 9: the contact force exchanged between the pantograph head and the test rig is measured. According to the value of measured contact force a real-time model of the catenary is able to calculate the corresponding displacement of the contact points. The calculated displacements are then imposed as imposed motion references that are reproduced by rig actuators. In the simplified scheme of Figure 9 both lateral and vertical motion of contact points are reproduced since test rigs able to simulate multiple degree of freedom exists as stated for example by (Baldauf, 2001). The simulation of tridimensional dynamics of the catenary is computationally intensive and also measurements and actuation systems needed to simulate a tridimensional motion of catenary are generally more expensive. As consequence in the most commonly diffused HIL application as the test rig of Politecnico di Milano (Facchin etti,2004),(Facchinetti,2009) the flexible model of the catenary is planar and only one degree of freedom, corresponding to the vertical motion of the pantograph head is actuated.

For a brief comparison of different modeling techniques applied to HIL testing of pantograph there are several interesting works in the literature as (Hedayati, 2010),(Arnold, 2000). Cross-Validation and calibration of simplified catenary models used for HIL testing is usually done by comparing the results obtained with more complete tri-dimensional code (Parka, 2003) often developed with the support of experimental data like the French OSCAR (Outil de Simulation du CAptage pour la Reconnaissance des défauts), (Massat, 2006). Also international standards offer criteria for the validation and verification (EN50318, 2002) of pantograph-catenary simulation codes.

The limited bandwidth of contact force measurements and the use of simplified catenary models involve a bandwidth limitation for HIL testing to about 20Hz. As a consequence, also for the actuation system a dynamical response corresponding to a cut-off frequency of 20-30 Hz is considered acceptable as visible in the graphics of Figure 10 (a), where typical specifications for the dimensioning of the actuator used to impose the vertical motion to pantograph head are shown. The evaluation of the actuator bandwidth must take in to account the compliances introduced by the modal response of the rig portal which has to be as stiff as possible considering that the lower frequency modes of the structures may be excited by actuator reaction forces as in the case study described by (Allotta, 2009) and shown in Figure 10 (b), where the first resonant mode of

Figure 9. Simplified scheme of a HIL rig for dynamical and modal testing of pantographs



the mechanical structure lays in the range between 20 and 30Hz.

Most commonly used actuators are hydraulic since they can be easily tuned to manage speed and position loops with a smooth friction-free behavior; also the use of linear motors has been explored and suggested in literature (Allotta, 2009).

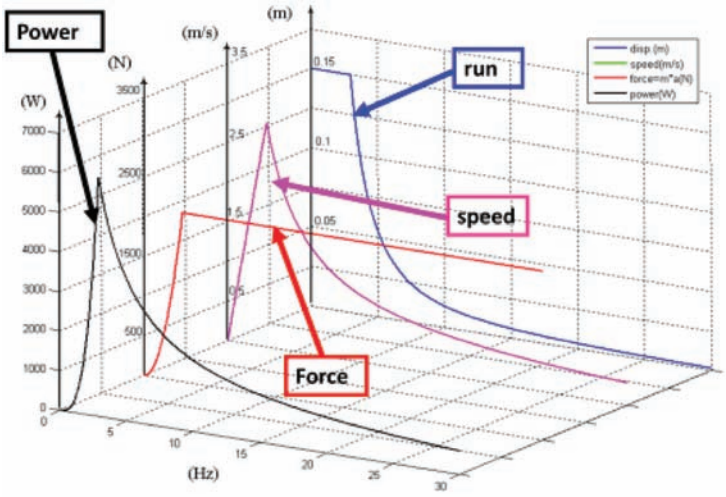
All the mentioned test rigs are often used in a complementary way in order to investigate different aspects of a complex problem reducing as much as possible extended experimental campaigns with high speed trains on railway lines. In fact this kind of tests are time consuming, expensive and potentially dangerous since a failure of the tested pantograph may damage the overhead line causing the unavailability of an expensive infrastructure with a potential perturbation also of the commercial traffic.

As a consequence the definition of optimization and qualification criteria of pantograph is still an open point for researchers considering the continuous increase of performances due to the wide diffusion and the technological improvements of high speed trains.

However optimal response of the pantograph mainly depends from a good mutual interaction with catenary: as a consequence dynamical behavior of the pantograph considering also non-secondary aspects due to available encumbrances, have to be shaped to respect specifications which mainly affects its mechanical impedance. Typical measurement of mechanical impedance are also prescribed by some national standards such as (RFI, 2008): typically the transfer function between the contact force applied on the pantograph head F_c and the corresponding motion of the head measured in terms of displacement x, speed $\dot{x}$ or more commonly acceleration $\ddot{x}$ are used in the range of 0-20Hz (11):

$$\text{Receptance } \alpha(\omega)=\frac{x(\omega)}{F_c(\omega)};\quad \text{Eq. or Apparent Stiffness } K_{eq}(\omega)=\frac{1}{\alpha(\omega)};$$
$$\text{Mobility } Y(\omega)=\frac{\dot{x}(\omega)}{F_c(\omega)};\qquad \text{Mech.Impedance } z(\omega)=\frac{1}{Y(\omega)};$$
$$\text{Inertance } I(\omega)=\frac{\ddot{x}(\omega)}{F_c(\omega)};\quad \text{Eq. or Apparent Mass } M_{eq}(\omega)=\frac{1}{I(\omega)};\qquad (11)$$

Figure 10. (a) An example of performance specifications for the dimensioning of the actuator used to impose vertical motion to the head of a tested pantograph (b) An example of structure for a test rig and the corresponding eigen-frequencies calculated with a FEM model (Allotta, 2009)



(a)



Mode N.	Frequency[Hz]
1	20.98
2	28.08
3	29.14
4	37.25
5	37.57
6	38.30
7	41.83
8	42.30
9	43.61

(b)

In particular according to (RFI, 2008), the maximum apparent mass $M_{eq}(\omega)$ measured in the range of 0-20Hz have to be limited under a known value which depends from the maximum operating speed of the tested pantograph. Mechanical impedance measurements are also used as indexes to evaluate the performances of innovative pantographs (Allotta, 2008),(Yamashita, 2011).

Bandwidth of force measurements it's necessary limited and also high frequency arcing is often associated to the concurrent action of different physical phenomena like aerodynamic disturbances, tribology of sliding surfaces, inductive load of line and connected power equipment.

As a consequence measurement and statistical evaluation of arcing is often used as a complementary test which is often verified on rigs devoted to the study of wear and arcing.

WEAR OF SLIDING SURFACES AND ARCING: APPLICATION OF HIL TECHNIQUES TO PANTOGRAPH TESTING

In order to test the wear of pantograph contact strips it is necessary to reproduce arcing which is responsible of a significant part of the damage of sliding surfaces. Also simulation of arcing is useful to evaluate high frequency losses of contact with durations of few milliseconds whose detection through contact force estimations is quite difficult or impossible. This high frequency losses are however very important since they produce wear, electrical disturbances on current absorbed by vehicle power devices and electromagnetic noise which may be introduce disturbances on communication and signaling systems. Arcing is usually detected in terms of measurements on collected current or sensors able to detect the ultra-violet emissions associated to sparks (Bruno, 2001),(Hayasaka, 2009).

There are several examples of rigs devoted to this kind of tests, however in this chapter the attention is concentrated on a possible layout described in the schemes of Figures 11 (a) and (b) which are inspired to existing rigs such as (Bucca, 2010),(Bucca, 2009), (RTRI, 2010).

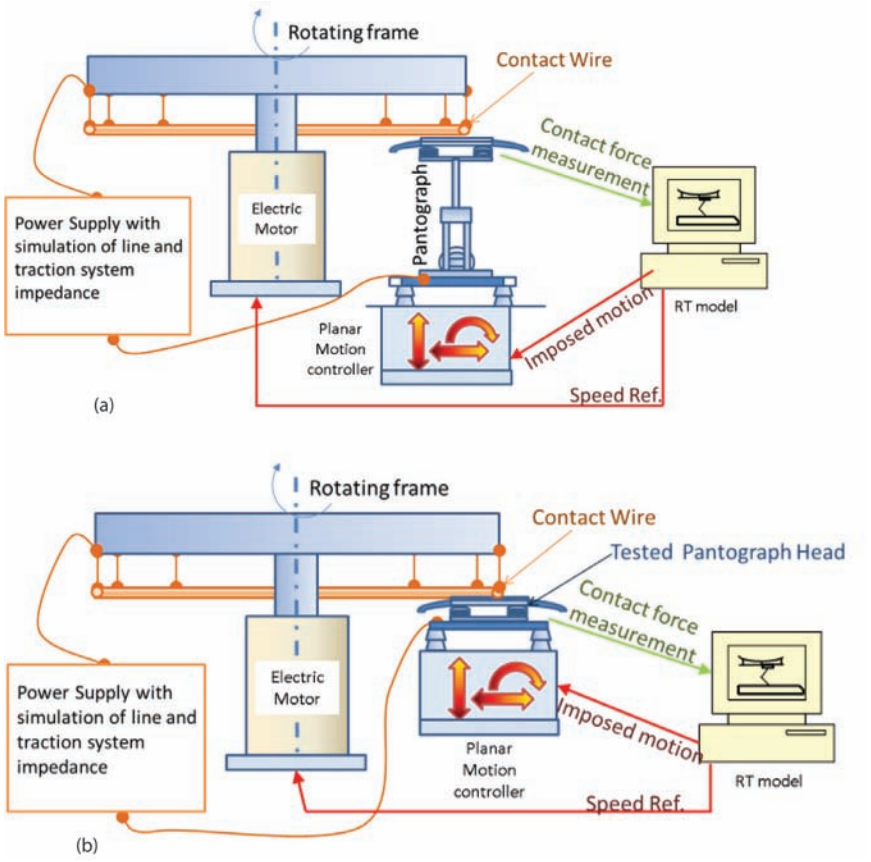
As visible in the schemes of Figures 11 (a) and (b), a typical rig is composed of a rotating frame on which is suspended a circular section of the contact copper wire of the simulated catenary. An electric motor is used to control the angular speed of the rotating frame in order to reproduce the longitudinal motion of the train. From an electrical point of view, the rotating contact wire and the pantograph are connected to an electrical power system able to reproduce the equivalent electrical impedances introduced by the overhead line and by the underlying electrical load of the simulated train, such as traction system and other auxiliary power devices. A partial reproduction of the air flow investing the pantograph is often introduced mainly for cooling purpose in order to obtain a good reproduction of the thermal condition affecting the sliding surfaces of contact strips and wires.

In order to obtain near realistic testing conditions, the radius of the circular sections has to be has bigger as possible, however there are some limitations due to encumbrances, costs and inertial/ loads that affect the rotating machinery limiting the range of feasible solutions.

Equilibration, stability and vibration of a rotating machinery with a diameter of several meters and a tangential speed of hundreds of km/h as the simulated vehicle run; so no other mechanical actuation is usually placed on this structure whose design is quite critical. Relative motions between pantograph head and simulated catenary are reproduce by imposing the motion of a mobile platform on which is placed the tested pantograph (scheme of Figure 11 (a)) or simply the pantograph head (scheme of Figure 11 (b)). Imposed motions can be a recorded pattern obtained from offline simulations or experimental data. In case of HIL testing, imposed motions are calculated by a real-time model of pantograph-catenary interactions. Calculations are performed according contact force measurements on the head of the tested pantograph.

This kind of application is a typical example of HIL system in which over bandwidth response must be carefully evaluated: arcing is a phenomena corresponding to mechanical vibrations of contact strips and wires which are far above the maximum bandwidth allowed by several limitations of the control loop as for example:

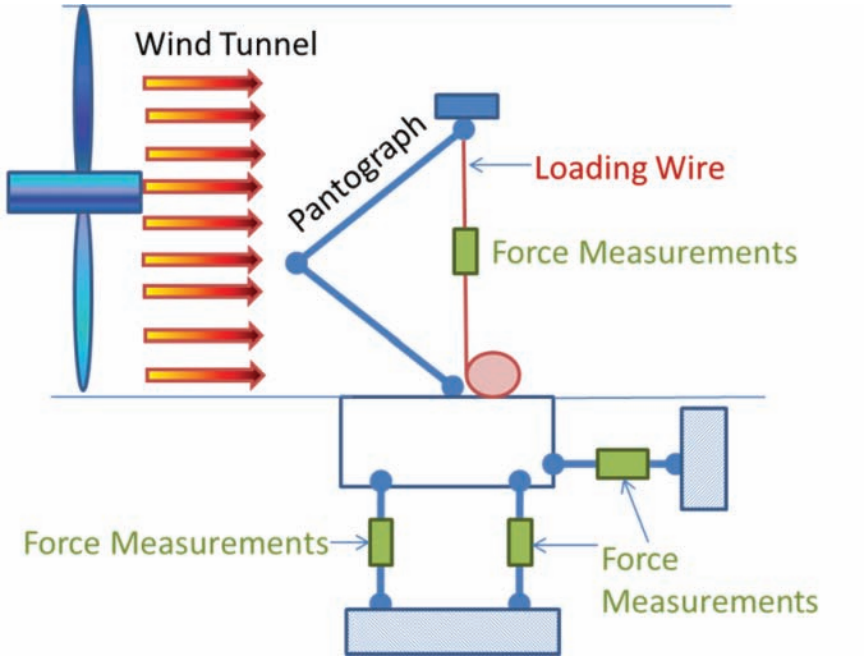
Figure 11. (a) Example 1 of rig for wear/arcing testing on pantographs (b) Example 2 of rig for wear/arcing testing on pantographs



- Actuation: required bandwidth, encumbrances and layout limitations and other environment related problems make technically difficult and expensive the design of actuation systems able to reproduce contact force fluctuations of 30÷50Hz for an HIL application.
- Contact force measurements: even considering the deformable behavior of contact strips, the bandwidth of contact force estimators is limited.
- Insufficient real-time model performance: computational resources and model validations are quite critical if it is required to model the pantograph-catenary interaction over a certain frequency limit especially when complex aero-elastic/aero-acoustic interactions are involved.

As a consequence of the above mentioned technical limits, at very high frequencies the simulation of the mechanical impedance of the catenary cannot be performed by the HIL control loop. High frequency response of the rig must be shaped in order to fit as much as possible the corresponding modal behavior and tribology of the tested system, so some details as the way in which the contact wire is constrained to the rig may be important for better simulation of high frequency arcing.

Figure 12. Wind tunnel tests, a typical layout



Wind Tunnel Test

Wind tunnel tests are performed not only to identify the aero-dynamical behavior of the pantograph but also to optimize aero-elastic and aero-acoustic interactions which may be difficult to investigate only with simulation models as stated by some interesting studies(Baldauf, 2001), (Kurita, 2010).

In Figure 12, a simplified scheme of a wind tunnel test is shown: the pantograph placed over a sensorized platform is subjected to the air flow generated by a wind tunnel; often a wire is used to preload the pantograph head. Aerodynamic forces globally applied on the pantograph are measured by load cells while accelerometers and deformation sensors such as traditional or Bragg optical strain gage are often used to evaluate vibrations induced by aero-elastic phenomena. Also acoustic microphone arrays are used to evaluate the corresponding noise emissions.

Typically this kind of rigs are not for HIL testing but they are useful in order to identify relevant pantograph properties. However some authors, as described in (Ikeda, 2005), are evaluating the application of active control systems on pantographs, based on the modulation/control of aerodynamic forces due to the interaction with the surrounding flow. The development of dynamical and modal HIL test rigs for pantographs has been boosted by studies and investigations on active suspension system. So it is possible in a medium-term future a wide application of HIL actuation systems also to this kind of rigs.

MUTUAL INTERACTION OF SAFETY-RELEVANT SUB-SYSTEMS WITH DEGRADED ADHESION CONDITIONS: THE APPLICATION OF HIL TESTING TO ODOMETRY,WSP, AND ANTI-SKID SYSTEMS

Neglecting the lateral dynamics of wheel-rail interaction, and assuming a rigid-body contact with (known) fixed rolling radius r_w of the wheel, the longitudinal behavior of an axle can be modeled

using the simplified planar model depicted in Figure 13 (a) whose dynamics is briefly described by Equation (12)

$$\begin{aligned} & m\ddot{x} = T_i - F_i - mg\sin(\alpha) \\ & P_i + mg\cos(\alpha) = N_i \\ & I_i\ddot{\theta}_i = M_{ei} - T_i r_w \\ & T_i = f_i\left(\Delta v_i^r, \Delta \dot{v}_i^r\right) N_i \quad \text{(adhesion law)} \end{aligned} \tag{12}$$

The tangential transmitted force T_i depends from wheel-rail adhesion factor $f_i\left(\Delta v_i^r, \Delta \dot{v}_i^r\right)$ where Δv_i^r is defined according (13):

$$\Delta v_i^r = \frac{\dot{x} - r_w \cdot \omega_i}{\dot{x}} \tag{13}$$

The typical behavior of the adhesion factor $f_i\left(\Delta v_i^r, \Delta \dot{v}_i^r\right)$ is visible in Figure 13 (b): in nominal adhesion conditions, the transmitted tangential force T_i is low with respect to the maximum transmissible value.

As visible in Figure 13 (b) the adhesion behavior suggested by models available in literature (Boiteux, 1986) implies a dependence of the adhesion law also from the presence of contaminants between the rolling surfaces of wheels and rails. Presence of contaminants drastically reduces and modify the shape of the adhesion law $f_i\left(\Delta v_i^r, \Delta \dot{v}_i^r\right)$ introducing further non-linear effects due to the chemical, thermal and mechanical interactions of the contaminants with the rolling surfaces: in fact slip and contact pressure between the rolling surfaces affect the mechanical dissipated energy and, consequently, thickness and properties of the contaminants, thus modifying the adhesion behavior. As consequence, degraded adhesion conditions are quite complex physical phenomena to reproduce.

In nominal adhesion conditions, pure rolling is a good approximation of wheel–rail interaction, so the relationship (14), between the longitudinal vehicle speed $\dot{x}$ and the angular speed of the i-th axle ω_i is valid assuming a known constant rolling radius r_w and negligible slide between rolling surfaces.

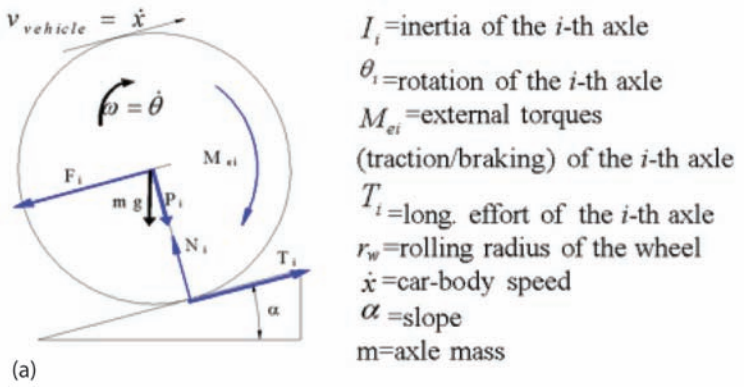
$$\dot{x} = \omega_i r_w \tag{14}$$

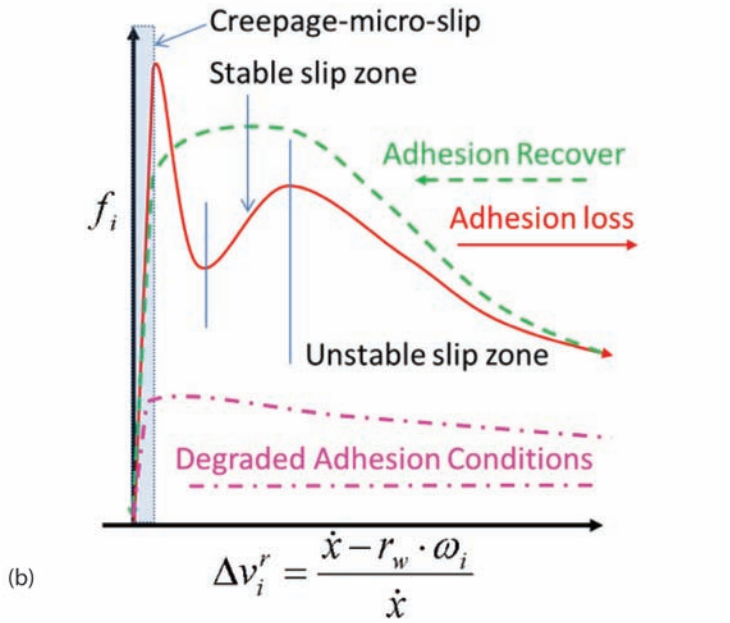
Tachometers, which are usually employed on rail vehicles in order to estimate the longitudinal speed of the train, work on this principle. If (14) is valid, the travelled distance *x* of the train along the line can be estimated with a simple operation of integration (15):

$$\begin{aligned} & x = \int_0^T \dot{x} dt \\ & \textit{where } T \textit{ is the observed period,} \\ & \textit{corresponding to the evaluated test run} \end{aligned} \tag{15}$$

Modern ATP (Automatic Train Protections), ATC (Automatic Train Control) and, more generally, signaling systems. need to estimate train speed and position along the line. The on-board sub-system devoted to the evaluation of train speed and position is called Odometry. The odometry function is safety-relevant since ATP-ATC has to decide the application of emergency braking procedures if the speed of the train is too high with respect to the available braking resources in order to assure the respect of a target maximum speed on a certain protected point/location over the line called "target" in the simplified scheme of Figure 14, taken from (Cocci, 2006). In particular the ATP-ATC system generates one or, more generally, a family of speed profiles, which represents the corresponding thresholds for the application of known safety procedure (alarms for the driver, traction cut off, emergency braking, etc.).

Figure 13. (a) Simplified planar wheel-rail model (b) Example of adhesion law



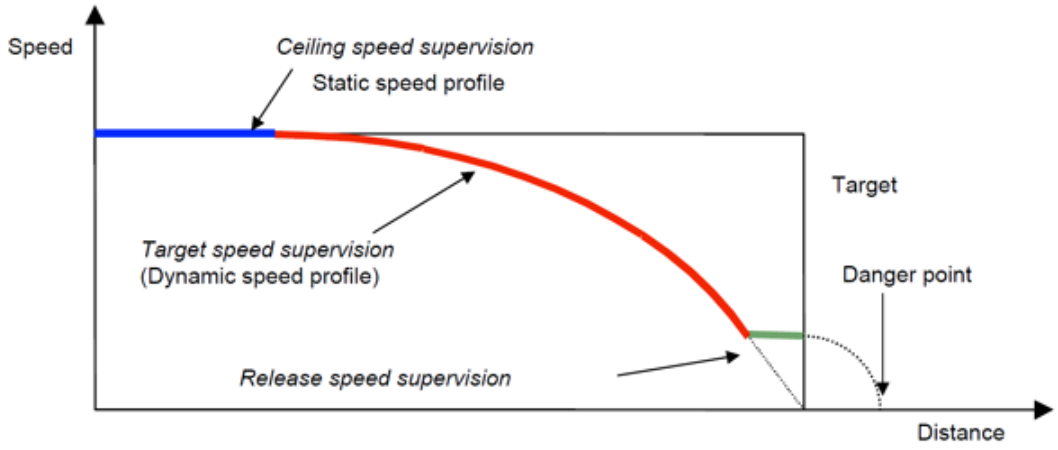


Unfortunately with degraded adhesion, prolonged sliding corresponding to high values of Δv_i^r occur.

Conventional odometry algorithms, like the Italian SCMT (Sistema Controllo Marcia Treno) (Trenitalia,2000), try to solve this problem using an estimation algorithm which is based on the redundant measurement of the angular velocities of two axles through GIT (Generatori di Impulsi Tachimetrici) usually optical, Wiegand, or Magneto-Resistive Sensors whose functionality may be assimilated to two channel incremental encoders. The estimation filters may be implemented with different data fusion techniques (Kalman filtering, crisp reasoning, fuzzy logic, neural networks, etc.).

However axle filtering criteria are deeply influenced by several external parameters which cannot be measured but only indirectly estimated, such as, for example, the application of braking/traction torques on axles which influence value and signs of the relative sliding of wheels. As a consequence, the filter has often a variable or adaptive criteria as a function of continuous or

Figure 14. Generation of a speed protection curve for an ATP/ATC system (Cocci,2006)



discrete states estimated from the signal analysis of GIT sensors. As example for a discrete state machine algorithm like Italian SCMT (Toni,2003) based on crisp criteria, evaluation scheme and parameters change according to the estimated maneuver applied to the vehicle/train such as traction, braking and so on. In this example, this choice reflects physical considerations since different maneuvers like braking or traction involve the application of different torques to axle and consequently a change of sign and values of axles relative sliding.

As a compromise between system availability and safety of the system some backup configurations/strategies corresponding to a degradation of GIT performances may be adopted. More generally for system safety the robustness of the system against undetected or partial performance degradation has to be carefully analyzed.

However even the performances of most advanced conventional GIT-based odometry algorithms are influenced by the shape of measured wheel speed profiles. Sowhen degraded adhesion occurs, the performances of a GIT-based odometry algorithm is influenced by the behavior of all the subsystem/physical effects which may potentially affect axle speed profiles and train longitudinal dynamics:

- **Wheelset and Mechanical Transmission Layout:** Two or more axles are mechanically constrained to rotate with the same angular speed, as for example by a gearbox. For example on many locomotives used both for commercial and freight transit an unique motor is used to actuate all the axles of the same bogie. On low speed or diesel vehicles the motion between all axles is generated by a single motor unit and distributed through an hydraulic/mechanical transmission system.
- **Electromechanical Interactions:** Often, even if two axles are completely independent each other from a mechanical point of view, their speed is in some way constrained: for example, two or more AC motors may be electrically connected in parallel to the same drive unit; Sothe same frequency and voltage are applied to both motors resulting in an electro-mechanical interaction between the two axles as described in (Allotta,2010).
- **WSP (Wheel Slide Protection System):** In order to prevent the locking of the wheels in the braking phase; the braking torque applied to each axle is controlled by the WSP system. WSP system accord-

ing regulation in force (UIC541-05,2005) is mainly composed of three components as visible in Figure 15 (a):

- GIT/tachometers: rotational speed of each axle is measured continuously
- E.C.U. (Electronic Control Unit): using GIT speed measurements, an electronic control unit is able to estimate a reference speed of the vehicle and the corresponding slips of wheels. According estimated slips values and their derivatives a control algorithm decides how to modulate the clamping pressure of brake cylinders.
- EVR-EVS, valves: a system of servovalves (usually 2 for each independently controlled axle), is used to regulate cylinders pressures. EVR-EVS valves can be modeled as a three way valve that can be used to connect the brake cylinder alternatively with the pressure supply coming from distributor or with the atmosphere.

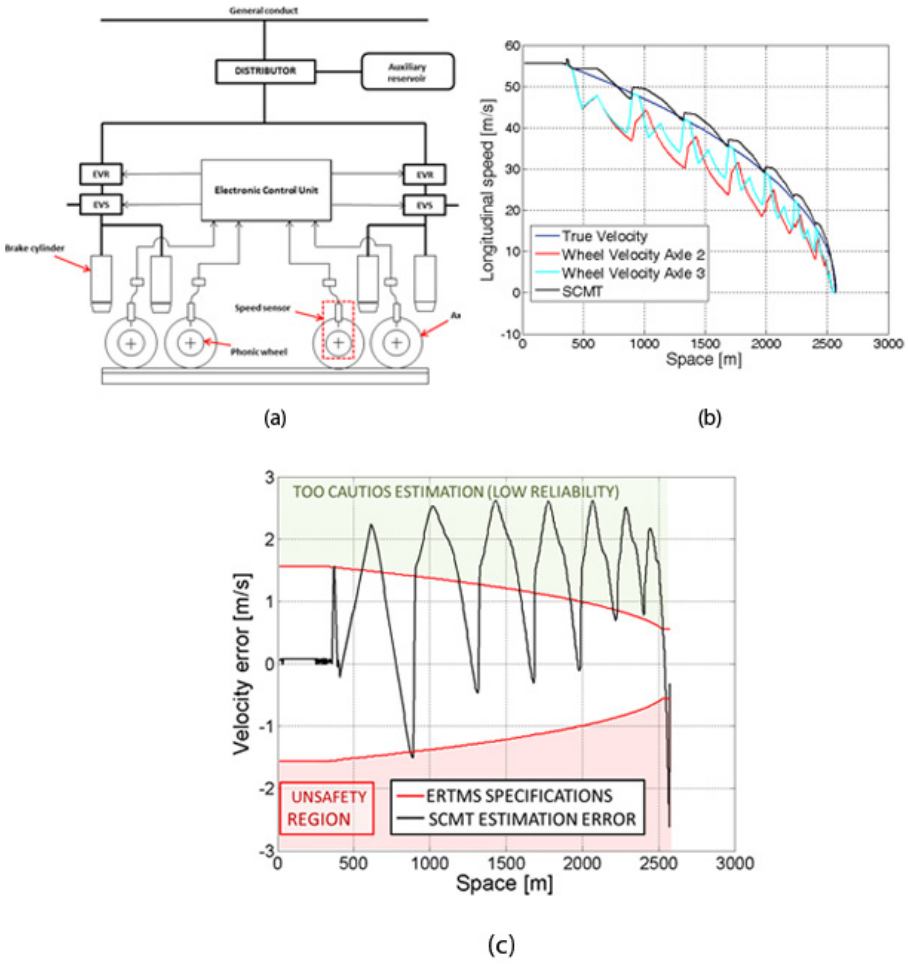
- **Traction Control System/Antiskid:** As well as the WSP controls the braking torques in order to prevent excessive wheel slip, the anti-skid system controls the torque references of traction motors, in order to prevent an excessive wheel skidding associated to a traction maneuver with degraded adhesion conditions.
- **Wheel-Rail Adhesion Features and Contaminants:** The often abused term "degraded adhesion condition," is used to depict a situation in which the wheel-rail adhesion factor is not sufficient to assure the transmission of tangential efforts between wheel and rail; however the real contact behavior when degraded adhesion condition occurs, is strongly influenced by the type of contaminants that produces degradation and often by the tribological properties of the surface with may be changed by prolonged sliding between wheel and rail. As a consequence, regulation in forces concerning degraded adhesion tests (UIC541-05, 2005) for WSP systems, clearly indicates the type of contaminants that have to be used. Also it is commonly accepted that during prolonged experimental campaigns the quantity of contaminants may be changed in order to obtain the same degraded adhesion conditions which is evaluated in terms of wheel-rail sliding.

As example in Figures 15 (b) and (c), some results concerning the simulated performances of SCMT odometry algorithms during a braking maneuver with degraded adhesion conditions are shown: in the example of Figure 15 (b) the WSP installed on the coach is able to regulate the brake effort avoiding the lock of the axle whose angular speed is used by SCMT algorithm to evaluate vehicle speed even with very degraded adhesion conditions. The corresponding speed estimated by SCMT is shown in Figure 15 (c) and compared with ERTMS specifications: considering the extremely degraded adhesion conditions both WSP and SCMT algorithm performed well, SCMT is not able to respect tolerances imposed by ERTMS specifications but the errors involve an overestimation of train speed so it is too cautious from a safety point of view since it produces an overestimation of the travelled distance.

However this example makes quite clear that the performances of odometry algorithms are deeply influenced by efficiency and behavior of the other vehicle subsystems cited above since the behavior of the estimation error Δv_{est} clearly resembles the corresponding axle speed measurements as visible in Figures 15 (b) and (c).

$$\Delta v_{est} = \dot{x}_{est} - \dot{x};$$
$$\dot{x}_{est} = \text{carbody speed estimated by odometry algorithm} \tag{16}$$

Figure 15. (a) WSP,simplified functional scheme (b) Simulation of degraded adhesion test, axle speed behavior and corresponding speed estimation (c) ERTMS specification and corresponding velocity error of the simulated SCMT algorithm



In particular the performances of GIT based algorithms are influenced by the statistical distribution of absolute wheel-rail sliding Δv_i^a which are defined according (17).

$$\Delta v_i^a = \dot{x} - r_w \cdot \omega_i \tag{17}$$

In particular, considering an algorithm based on the measurement of two axles' angular velocities, the following statistical quantities approximately describe the quality of measured profiles in terms of odometry expected performance:

Mean axle sliding Δv_m^a : the arithmetic mean (over a given time interval) of both Δv_i^a vectors describes how measured speed profiles differ from vehicle speed; obviously, with nominal adhesion conditions Δv_m^a is near to zero assuming that, in the observed samples, the maneuver is known and homogeneous (only braking or only traction, etc.)

$$\Delta v_m^a = \frac{1}{2T}\int_0^T \left(\Delta v_2^a + \Delta v_3^a\right) dt$$
$i = 2,3$ indexes,
corresponding to axles acquired by the odometry algorithm
(18)

Standard deviation of axle profiles Δv_{std}^a: standard deviation calculated on both Δv_i^a vectors are a good index of the variability of sliding vectors respect to the mean value Δv_m^a: for a known value of Δv_m^a an higher value of Δv_{std}^a involve an higher variability of axle speed profiles, which usually corresponds to a richer information content for the odometry algorithm and consequently to better performances.

$$\Delta v_{std}^a = \sqrt{\frac{1}{T}\int_0^T \left(\frac{\Delta v_2^a + \Delta v_3^a}{2} - \Delta v_m^a\right)^2 dt}$$
(19)

Axle Cross-Correlation factor Δv_{xy}^a: Δv_{xy}^a, defined according, is the normalized cross-correlation factor between the sliding of both axles observed by GIT sensors. In case of degraded adhesion conditions high values of Δv_{xy}^a (as example $\Delta v_{xy}^a = 1$)involve that the observed axles has almost the same information content so the redundant acquisition of two axles is useless for the odometry algorithm. As a consequence, in case of degraded adhesion, corresponding to high modulus of Δv_m^a, it is preferable to have low values of Δv_{xy}^a.

$$\Delta v_{xy}^a = \frac{C(2,3)}{\sqrt{C(2,2)C(3,3)}}$$
where $C(i,j) = \frac{1}{T}\int_0^T \left(\left(\Delta v_i^a - \mu_i\right)\left(\Delta v_j^a - \mu_j\right)\right) dt$; $\mu_i = \frac{1}{T}\int_0^T \Delta v_i^a dt$;
(20)

Both measured and simulated axle speed profiles are affected by high frequency noise whose spectral content is usually far above the bandwidth of the on-board subsystem: as a consequence. speed profiles used to evaluate statistical index described by Equations (17),(18),(19),(20) are pre-filtered with a low pass filter (typically a 3rd or higher order, Butterworth filter) with a cut-off frequency of about 20Hz, this is a very cautious bandwidth, considering that the typical task/sampling frequency at which are implemented many odometry algorithms lays in the range of 5-20Hz (typical 10Hz). Also the dynamical behavior of many on-board systems which influence axle behavior such as pneumatic brake actuators is limited to few Hz.

Performance evaluation of an odometry algorithm is usually defined with respect to specifications provided by widely known standards such as ERTMS specifications in terms of error on estimated speed and position. Also simplified numeric indexes may be proposed in order to compare performances of different odometry algorithms Several Indexes, mainly based on statistical operators applied on speed and position estimation errors may be proposed, such as the mean speed error $\Delta \boldsymbol{v}_{e\_c}$ with respect to a family of tolerance curves $\boldsymbol{v}_c$ which may be a function of speed as for example the ERTMS velocity error specifications in Box 3.

Also as visible in Figure 15 (a). (b). and (c) a positive estimation error Δv_{est} is preferable for safety reasons so further indications concerning the statistical occurrence of the sign of the estimation error may be useful for a more sophisticated analysis (22).

$$\Delta v_{sign\%} = 100\,\frac{\int_0^T \left(\Delta v_{es} - v_{ERTMS} \geq 0\right) dt}{\int_0^T \left(\left|\Delta v_{es}\right| \geq \left|v_{ERTMS}\right|\right) dt}$$
(22)

In particular in Figures 16 (a) and (b), a long/ composite mission profiles with several traction, coasting and braking maneuvers are shown, where

Box 3.

$$\Delta v_{e\_ERTMS} = \frac{1}{x}\int_0^T \left(\Delta v_{est} - v_{ERTMS}\right) dt = \frac{1}{x}\int_0^T \left(\dot{x}_{est} - \dot{x} - v_{ERTMS}\right) dt$$

$$v_{ERTMS} = \begin{cases} \dot{x}_{est} - \dot{x} \geq 0 \Rightarrow v^{+}_{ERTMS}\left(\dot{x}\right) \\ \dot{x}_{est} - \dot{x} < 0 \Rightarrow v^{-}_{ERTMS}\left(\dot{x}\right) \end{cases} \quad (21)$$

where: $v^{+}_{ERTMS}\left(\dot{x}\right), v^{-}_{ERTMS}\left(\dot{x}\right)$ are the curves corresponding to speed error performances on estimation according ERTMS specifications

a vehicle with two bogies and four independent axles (B_0-B_0 wheel set) is considered: in the first case, corresponding to figure 16/a, the two intermediate axles as prescribed by SCMT specifications are used as input for the SCMT odometry algorithm.

In the second case the same simulation is repeated considering as input for the SCMT odometry algorithm axle 2 and 4 corresponding to the rear wheels of each bogie.

The corresponding odometry estimation errors Δv_{est} for the two example introduced in Figures 16 (a) and (b) are shown in Figures 17 (a) and (b).

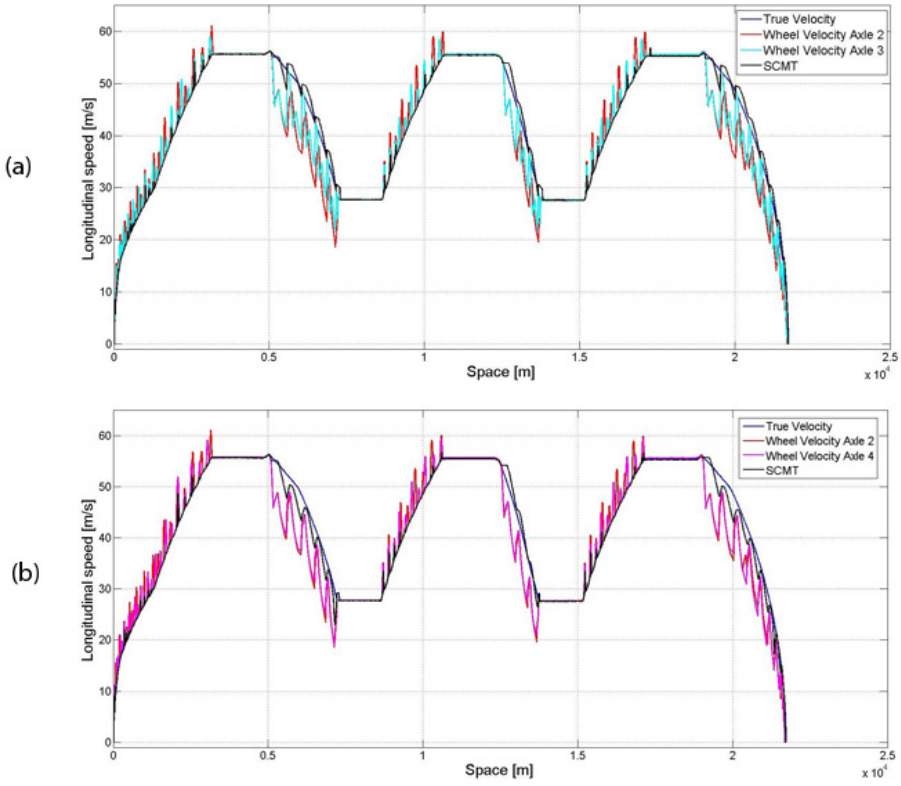
In Table 2 some results concerning input quality and performance indexes applied to the examples cited above are shown: it is interesting to notice that even considering long mission profiles with mixed applied maneuvers the proposed input and performance quality indexes are able to coherently describe the different quality of the input and the corresponding to different performances of the SCMT odometry algorithm.

Simplified indexes proposed in (17),(18),(19),(20),(21),(22), may be quite useful to reduce as much as possible the population of different tests that have to be performed. However, in order to assure safety and reliability of the odometry systems, thousands of different experimental tests have to be performed in order to investigate the response of the system among the range of possible working conditions

For this reason, in order to reduce cost and time consumptions HIL testing may be useful. A typical example is the Trenitalia test rig of Firenze Romito, also known as the "MI-6" test rig," which was originally developed for the virtual testing of Italian SCMT odometry algorithm (Toni,2003) and whose layout for odometry testing is visible in Figure 18. A real-time model of the train is used to calculate axle speed profiles considering different operating conditions. In order to reduce the needed computational effort, the model of the train is planar (lateral dynamics neglected). Axle speeds calculated by the real-time model are physically reproduced by velocity-controlled actuators which drive the corresponding GIT sensors of the tested odometry board. Commands from the tested ATP/ATC system can be used to modify/influence the simulated train behavior. In the original version of the test rig also the longitudinal relative speed between car-body and rails was reproduced by a system of belts in order to extend the capability of the rig to the testing of odometry systems which use absolute speed sensors like radar-doppler (Corbrion,2001).

Since the HIL testing of WSP system is a quite diffused practice, the rig was also equipped with a modular array of pneumatic components including brake cylinders and valves in order to simulate the braking plant of a railway vehicle: in this way the same test rig can be used also for the HIL testing of WSP boards which is a quite diffused

Figure 16. (a) Composite mission of profile with degraded adhesion conditions considering as input the speed measurements of the intermediate axle speed of a vehicle with B0-B0 wheelset (b) Composite mission of profile with degraded adhesion conditions considering as input the speed measurements of the rear axle speed of a vehicle with B_0-B_0 wheelset



application of HIL testing widely adopted by different research and industrial groups as in the examples (Guglielmino, 2006),(Faiveley, 2000) since the use of this kind of rigs is admitted and regulated also by UIC regulations (UIC541-5, 2005).

The modular structure of the rig and of its control software makes easy to perform also individual/customized test of individual components like GIT sensors.

Modularity and Flexibility are indispensable requisites for this kind of applications since there is a wide variety of different components with a continuous technological evolutions that has to be tested on the rig in order to assure the long term economical sustainability of this kind of laboratories.

Also an open architecture is needed in order assure the possibility to modify the testing layout in order to adapt the rig to the testing of innovative and/or customized products.

In particular especially when the tested systems perform the data fusion of different kinds of signals, all the synthesized signals must be coherent each other according to tested scenario: test rig limitations in terms of the bandwidth within which it is assured a correct emulation of the desired dynamics is usually constrained by the less performing components. As a consequence, a detailed and accurate engineering design is often

Figure 17. (a) Composite mission of profile with degraded adhesion conditions considering as input the speed measurements of the intermediate axle speed of a vehicle with B_0-B_0 wheelset (b) Composite mission of profile with degraded adhesion conditions considering as input the speed measurements of the rear axle speed of a vehicle with B_0-B_0 wheelset

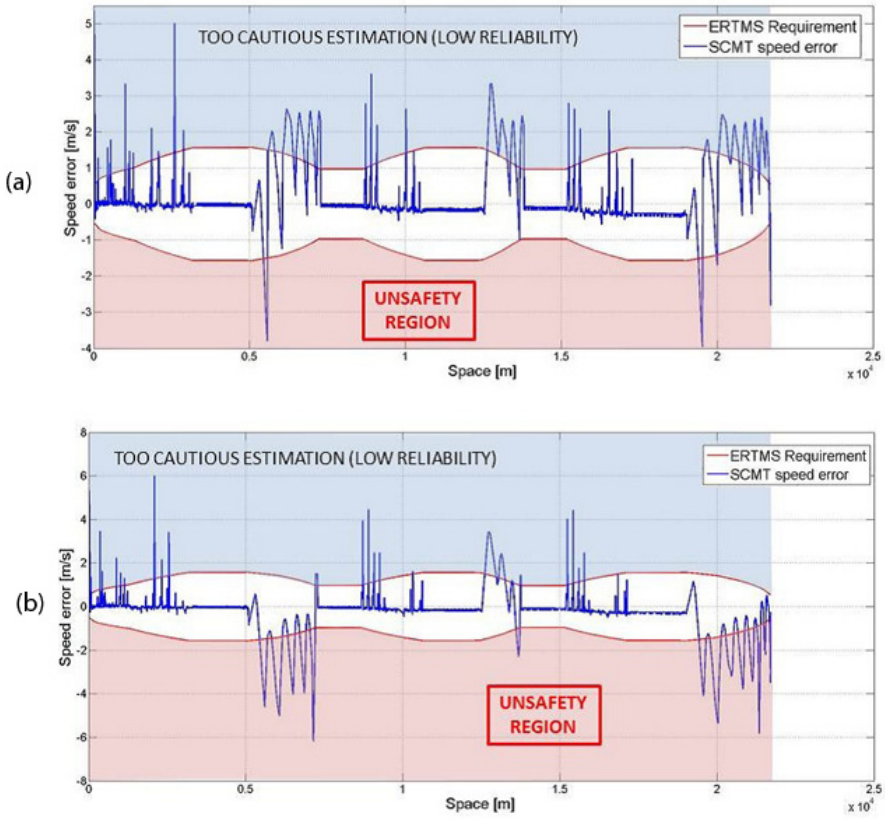


Table 2. Comparison of different input quality and performance indexes applied to the two examples described in Figures 16 (a) and (b)

SCMT Input	$\Delta v_m^a \left[m/s\right]$	$\Delta v_{std}^a \left[m/s\right]$	Δv_{xy}^a	Δv_{e_ERTMS}	$\Delta v_{sign\%}$
Axles 2 and 3	1.42	3.41	0.80	0.4	88.5%
Axles 2 and 4	1.65	3.86	0.92	-0.44	25%

Figure 18. Layout of Trenitalia MI-6 rig for HIL testing of WSP and odometry boards (Pugi,2006)

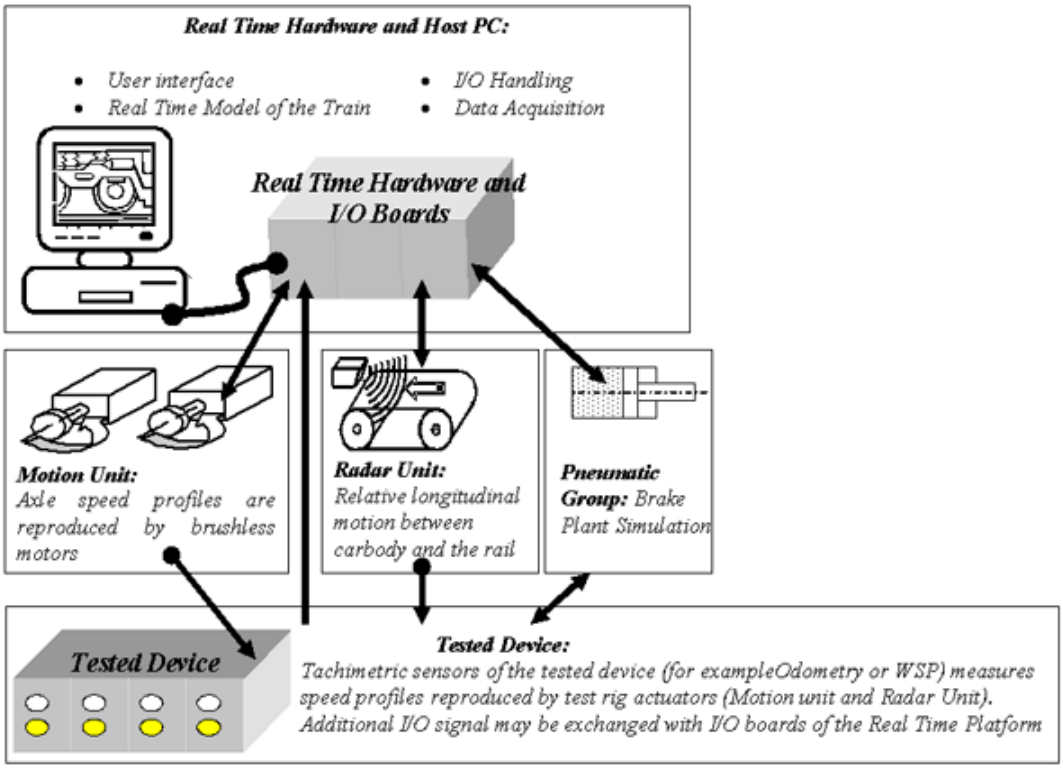
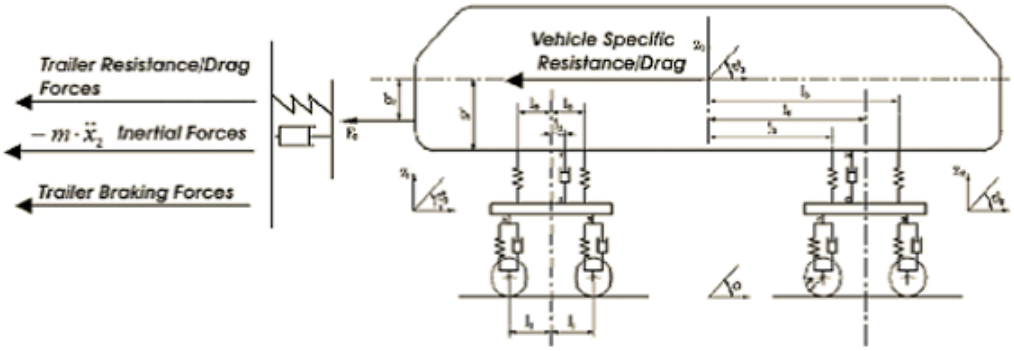


Figure 19. Example of vehicle planar model used for HIL simulation of Odometry and WSP systems (Pugi,2006)



needed to assure balanced performances of all the involved components.

From a computational point of view most of the existing test rigs used for the HIL simulation simplified vehicles models, planar as the example in Figure 19, neglecting lateral dynamics. However the availability of increasing computational power and the developments of efficient vehicle and wheel-rail contact models optimized for distributed computing and for real-time implementation makes possible further increase in the level of accuracy and complexity of this kind of HIL testing applications (Meli,2008), (Pugi, 2011).

Further improvements may also involve the HIL simulations of degraded adhesion conditions to a whole vehicle using scaled or full scale roller rigs (Malvezzi, 2008), (Allotta, 2010) as in the innovative test rig built in Italy in the new testing facility of Firenze Osmannoro.

CONCLUSION AND FUTURE DEVELOPMENTS

A modern railway system is composed by several on boards components and subsystems which involve a different engineering topics and competences. Also HIL testing is a flexible and almost mature approach that can be extensively applied to most of the components of a vehicle.

Consequence of this wide number of applications is the need for multidisciplinary system designers able to understand critical aspects involved in tested component or subsystem and to coordinate a team of specialists more related to particular aspects of the project.

For multidisciplinary nature of HIL testing different research topics ranging from smart actuators and sensors to improvements of computational resources should be suggested to the scientific community.

However considering the increasing performances in terms of real-time simulation models and timing, optimization of computational resources (automatic code parallelization, efficient computational methods, efficient communication and interfacing etc.) is still fundamental for the development of most demanding applications.

REFERENCES

Allotta, B., Pugi, L., & Bartolini, F. (2008, October). Design and Experimental Results of an Active Suspension System for a High-Speed Pantograph. *IEEE/ASME Transactions on Mechatronics*, *13*(5). doi:10.1109/TMECH.2008.2002145

Allotta, B., Pugi, L., & Bartolini, F. (2009a) "An active suspension system for railway pantographs: the T2006 prototype," *Proceedings of the IMechE, Part F: Journal of Rail and Rapit Transit, 223(* pp. 15-29)

Allotta, B., Pugi, L., & Bartolini, F. (2009b) Design and Testing of Innovative Pantographs: a general overview, *Proceedings of the International Seminar-Workshop on Power Transmission in High Speed Railway Systems*, Amiens, France, 4th December 2009

Allotta, B., Pugi, L., & Bartolini, F. (2010) *Mutual interaction of parallel connected induction motors on degraded adhesion conditions*, Proceedings of the 1st Joint International Conference on Multibody System Dynamics May 25-27, 2010, Lappeenranta, Finland

Allotta, B., Pugi,L., Malvezzi, M., Bartolini, F.,& Cangioli, F. (2010) A scaled roller test rig for high-speed vehicles, *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility48*(1) 3 – 18

Arnold, M., & Simeon, B. (2000). Pantograph and catenary dynamics: a benchmark problem and its numerical solution. *Journal of Applied Numerical Mathematics*, *34*(4), 345–362. doi:10.1016/S0168-9274(99)00038-0

Baldauf, W., Blaschko, R., Behr, W., Heine, C., & Kolbe, M. (2001) Development of an actively controlled, acoustically optimised single arm pantograph, *Proceedings of the World Congress of Railway Research WCRR 2001*, Cologne.

Boiteux, M. (1986) Le problème de l'adèrence en freinage [The problem of adhesion in braking], *Revue gènèrale des chemins de fer*, [General review of the railways] (pp. 59–72.) Fèvrier

Bruno, O., Landi, A., Papi, M., & Sani, L. (2001). Phototube sensor for monitoring the quality of current collection on overhead electrified railway. *Proceedings of the Institution of Mechanical Engineers. Part F, Journal of Rail and Rapid Transit*, *215*(3), 231–241. doi:10.1243/0954409011531549

Bucca, G., & Collina, A. (2009). A procedure for the wear prediction of collector strip and contact wire in pantograph-catenary system. *Wear*, *266*(1-2), 46–59. doi:10.1016/j.wear.2008.05.006

Bucca, G., Collina, A., Manigrasso, R., Mapelli, F., & Tarsitano, D. (2010). Methodology for correlating the quality of the pantograph-catenary contact with the harmonic content of the current collected. A case of multiple current collection. *Ingegneria Ferroviaria.*, *3*, 211–237.

CENELEC-EN 50206-1 (2010) *Railway applications - Rolling stock - Pantographs: Characteristics and tests - Part 1: Pantographs for main line vehicles* CENELEC

CENELEC EN 50367 (May 2010) *Railway applications - Current collection systems - Technical criteria for the interaction between pantograph and overhead line (to achieve free access) - Incorporating corrigendum*

Cocci, G., Malvezzi, M., Palazzolo, A., Presciani, P., Pugi, L., & Violani, M. (2006) *Braking Performance Monitoring in Service for the Validation of the Safety Margins used for the Definition of Braking Curves of ATP/ATC Systems*, World Congress on Railway Research 5-7June 2006 Montreal

Corbrion, C., T. Ditchi, T., Hole, S., Carreel, E., & Lewiner, J.,(2001) A broad beam Doppler speed sensor for automotive applications, *MCB University Press Sensor Review 21*(1). 28-32

Directive (2009*) 2008/57/EC on the interoperability of the rail system within the Community Technical Specification for Interoperability*, version EN02 del 17.12.2009

ERRI B 126/RP 18 (2000) *Braking Problems, Dynamometers for internal approval of friction materials*.

A. Facchinetti, F. Fossati, F. Resta, and A. Collina, (2004) Hardware in the loop test-rig for identification and control application on high speed pantographs, *Shock Vib11*(3/4),445–456.

Facchinetti, A., & Mauri, M. (2009). Hardware-in-the-Loop Overhead Line Emulator for Active Pantograph Testing. *IEEE Transactions on Industrial Electronics*, *56*(10). doi:10.1109/TIE.2009.2023632

Faiveley (2000) *Anti-skid system ANG 06/98 30/08/00 12:2* Datasheet retrieved from: http://www.faiveley.com/uk

Guglielmino, E. (November, 2004) Flexible Waveform Generation Accomplishes Safe Braking, *Evaluation Engineering*, Retrieved from: http://www.evaluationengineering.com

Hayasaka, T., Shimizu, M., & Nezu, K. (2009). Development of Contact-Loss Measuring System Using Ultraviolet Ray Detection Development of Contact-Loss Measuring System Using Ultraviolet Ray Detection. *Quarterly Report of*, *RTRI50*(3), 131–136.

Ikeda, M., Suzuki, M., & Yoshida, K. (2005) Application of Jet Ejection to control Contact Force of Pantographs for High Speed Trains *Proc. of the 6th Symposium on Smart Control of Turbulence*, Tokio 06-09-2005

Images and technical documentation available at the official site of Railway Technical Research Institute http://www.rtri.or.jp/rtri/facility2_E.html

Kia, S. H., Bartolini, F., Mpanda-Mabwe, A., & Ceschi, R. (2010), Pantograph-catenary interaction model comparison, *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, (pp.1584-1589), Washington, DC: IEEE Press.

Kurita, T,. Hara, M., Yamada, H., Wakabayashi, Y., Mizushima, F., Satoh, H., & Shikama, T. (2010) Reduction of Pantograph Noise of High-Speed Trains, *Journal of Mechanical Systems for Transportation and Logistics*, *3*(1) Special issue on STECH'09.63-74

Malvezzi, M., Allotta, B., & Pugi, L. (2008a). Feasibility of Degraded Adhesion Tests in a Locomotive Roller Rig, *Proc. of the IMechE. Journal of Rail and Rapit Transit*, *222*(1).

Malvezzi, M., Allotta, B., & Pugi, L. (2008b) Feasibility of Degraded Adhesion Tests in a Locomotive Roller Rig, *Proceedings of the IMechE, Part F: Journal of Rail and Rapit Transit, 222* (1) Part F; 27-43, ISSN: 0954-4097.

Massat, J. P., Bobillot, A., & Laine, J. P. (2006) Robust Methods for Detecting Defects in Overhead Contact Line Based on Simulation Results, *Proceedings of III European Conference on Computational Mechanics 2006,* The Netherlands: Springer - 978-1-4020-5370-2

Meli, E., Malvezzi, M., Papini, S., Pugi, L., Rinchi, M., & Rindi, A. (2008). A railway vehicle multibody model for real-time applications. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, *46*(12), 1083–1105. doi:10.1080/00423110701790756

Mpanda, A. (2009)Real-Time Test Rig & HIL Simulation Platform for Testing Pantograph-Catenary Interaction, *Proceedings of the International Seminar-Workshop on Power Transmission in High Speed Railway Systems*, Amiens, France

Parka, T. J., Han, C. S., & Jan, J. H. (2003). Dynamic sensitivity analysis for the pantograph of a high-speed rail vehicle. *Journal of Sound and Vibration*, *266*, 235–26. doi:10.1016/S0022-460X(02)01280-4

Pugi, L., Malvezzi, M., & Tarasconi, A. Palazzolo, A., Cocci, G, & Violani, M. (2006)Simulation of WSP Systems on MI-6 Test Rig, *Vehicle system and dynamics* (Taylor and Francis) *44*, 843-852, ISBN 978-0-415-43616-8

Pugi, L., Ridolfi, A., Allotta, B., Malvezzi, M., Vettori, G., Cuppini, F., & Salotti, F. (2011)*A 3D Simulation Model of Train Dynamics for Testing Odometry Algorithms,* Proceedings of WCRR 2011 (World Congress on Railway Research), Lille,France 22-26 May 2011

Pugi, L., & Rinchi, M. (2002) test Rig for Train Brakes, *3rd AIMETA International Tribology Conference*, AITC, Salerno - Italy 18 - 20 September 2002.

Railway Applications (2002) *Current Collection Systems—Validation of Simulation of the Dynamic Interaction Between Pantograph and Overhead Contact Line, European Standard EN 50318*, Jul. 2002.

RFI-DTC/DNS/EE.ST TE 74 D (February, 2008), *Prove da eseguire per caratterizzazione di un pantografo a 3kV CC* [Tests to be performed for the characterization of a pantograph 3kV DC], Sciavicco, L., and Siciliano, B. (1996) *Modelling and Control of Robot Manipulators*.New York: McGraw-Hill.

Slotine, J. J., & Li, W. (1991). *Applied Non Linear Control*. Prentice Hall.

Toni, P., Malvezzi, M., Pugi, L., Rinchi, M., & Presciani, P. (2003)"Sviluppo e validazione di algoritmi di odometria per sistemi di controllo e monitoraggio ferroviari," [Development and validation of algorithms for odometry systems control and monitoring rail] *Ingegneria Ferroviaria* [Railway Engineering] 433/457 (in Italian).

Trenitalia S.p.A., (2000) SCMT, progetto dell'algoritmo per il calcolo della velocità istantanea del treno e dello spazio percorso, [SCMT, algorithmic project for the calculation of the instantaneous speed of the train and the distance traveled], *Unità Tecnologie Materiale Rotabile*, UTMR.DT.PS.31-10-2000

UIC541-3 (2010) *Brakes - Disc brakes and disc brake linings, edition(7th ed.)* UIC

UIC 541-05 (2005) *Brakes - Specifications for the construction of various brake parts - Wheel Slide Protection device (WSP)* (2nd ed.), November 2005 – Translation List of recent publications1/06 (date of issue 1/02/2006)ISBN2-7461-0969-7

UIC 541-4 (May, 2007) *Brakes - Brakes with composition brake blocks - General conditions for certification of composite brake blocks* (3rd ed)UIC

Yamashita, Y., Mitsuru, I., Tatsuya, K., Arata, M., Daisuke, I., & Kazusaku, F. (2011) *Advanced active control of a contact force between a pantograph and a catenary for a high-speed train,* Procedings of the 9th World Congress of Railway Research, May 22-26 May 2011

Section 5

Formal Methods

Chapter 12
The Role of Formal Methods in Software Development for Railway Applications

Alessandro Fantechi
Università degli Studi di Firenze, Italy

ABSTRACT

Formal methods for thirty years have promised to be the solution for the safety certification headaches of railway software designers. This chapter looks at the current industrial application of formal methods in the railway domain. After a recall of the dawning of formal methods in this domain, recent trends are presented that focus in particular on formal verification by means of model checking engines, with its potential and limitations. The paper ends with a perspective into the next future, in which formal methods will be expected to pervade in more respects the production of railway software and systems.

INTRODUCTION

The challenges posed by the new scenarios of railway transportation (liberalization, distinction between infrastructure and operation, high speed, European interoperability, ...) have a dramatic impact on the safety issues. This impact is counterbalanced by a growing adoption of innovative signaling equipments (most notable example is ERTMS/ETCS) and monitoring systems (such as on board and wayside diagnosis systems). Each one of these devices include some software, which in the end makes up the major part of their design costs; the malleability of software is paramount for the innovation of solutions. On the other hand, it is notorious how software is often plagued by bugs that may threaten its correct functioning: how can the high safety standards assumed as normal practice in railway operation be compatible with such threats?

DOI: 10.4018/978-1-4666-1643-1.ch012

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

The employment of very stable technology and the quest for the highest possible guarantees have been key aspects in the adoption of computer-controlled equipment in railway applications. Formal proof, or verification, of safety is therefore seen as a necessity.

This chapter reviews current experiences and future trends in the application of formal methods in the railway area: after a recall of the first steps of formal methods in this domain, recent trends are presented, both from the point of view of safety guidelines, and from that of the practical applications, pointing to the most adopted techniques, in particular related to formal verification by model checking. The specific application to railway signaling equipment is dealt with some detail, and future trends will emerge from such discussion.

BACKGROUND

Early Applications of Formal Methods

Nowadays, the necessity of formal methods as an essential step in the design process of industrial safety-critical systems is widely recognized.

In its more general definition, the term formal methods encompasses all notations having a precise mathematical semantics, together with their associated analysis and development methods, that allow to describe and reason about the behaviour and functionality of a system in a formal manner, with the aim to produce an implementation of the system that is provably free from defects.

Railway signaling has been traditionally considered as one of the most fruitful areas of intervention for formal methods (Fantechi, Fokkink, & Morzenti, 2011). Already in the early nineties, a series of railway signaling products have benefited from the application of the B formal method in the design process.

The B method (Abrial, 1996) targets software development from specification through refinement, down to implementation and automatic code generation, with formal verification at each refinement step: writing and refining a specification produces a series of *proof obligations* that need to be discharged by formal proofs. The B method is accompanied by support tools, which include tools for the derivation of proof obligations, theorem provers, and code generation tools.

The B method has been successfully applied to several railway signalingsignaling systems. The SACEM system for the control of a line of Paris RER (DaSilva, Dehbonei, & Mejia, 1993) is the first acclaimed industrial application of B. B has been adopted for many later designs of similar systems by Matra (now absorbed by Siemens). One of the most striking application has been the Paris automatic metro line 14. The paper (Behm, Benoit, Faivre, & Meynadier, 1999) on this application of B reports that several errors were found and corrected during proof activities conducted at the specification and refinement stages. By contrast, no further bugs were detected by the various testing activities that followed the B development.

CENELEC Guidelines

The success of B has had a major impact in the sector of railway signaling by influencing the definition of the EN50128 guidelines (CENELEC, 2001), issued by the European Committee for Electrotechnical Standardization (CENELEC). These guidelines address the development of ”Software for Railway Control and Protection Systems”, and constitute the main reference for railway signaling equipment manufacturers in Europe, with their use spreading to the other continents and to other sectors of the railway (and other safety-related) industry.

The EN50128 document is part of a group of documents regarding the safety of railway control and protection systems, in which the key concept of Safety Integrity Level (SIL) is defined, a number ranging from 0 to 4, where 4 indicates a high criticality, 0 gives no safety concern. The SIL is actually a property of the system, related to the damage a failure of the system can produce, and is usually apportioned to subsystems and functions at system level in the preliminary risk assessment process. Also software functions are associated a level (Software SIL); assigning different SILs to different components helps to concentrate the efforts (and therefore the production costs) on the critical components.

The EN50128 guidelines however dictate neither a precise development methodology for software, nor any particular programming technique, but classify a wide range of commonly adopted techniques in terms of a rating with respect to the established SIL of the component.

Formal methods are rated as highly recommended for the software requirements specification and software design of systems/components with the higher levels of SIL. Formal proof is also highly recommended as a verification activity. The norm however does not dictate any process in which formal methods take a role, but just gives a list of the most common formal and semi-formal methods. Moreover, other combinations of highly recommended techniques, not including formal methods can be chosen: for example, testing combined to full traceability to requirements is a compliant, commonly used, approach to software verification of highest SIL software components.

The need for a revision of CENELEC norms has brought to a proposal by the TC9X/SC9XA Technical Committee, which has just been published, and which includes many more references to formal modelling and formal verification techniques, including the emergent model checking and abstract interpretation techniques, discussed later in this chapter.

RECENT ADVANCES

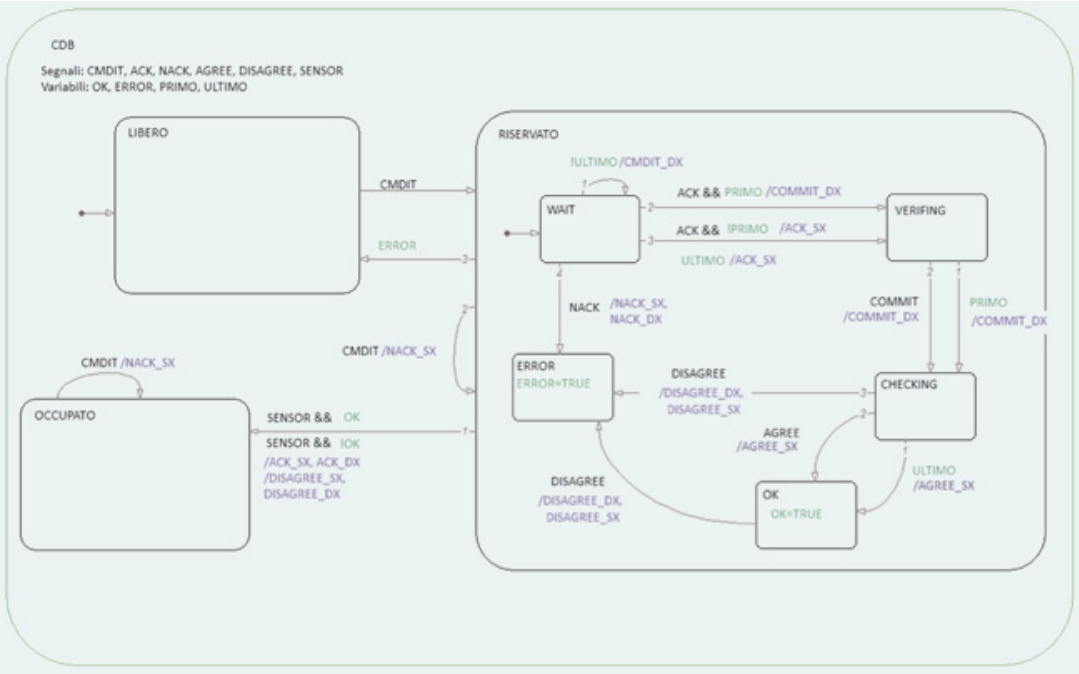
Indeed, methods like B or Z, which are still applied nowadays by the companies that have adopted them in their heydays - see for example (Lecomte, 2008), end up to require a substantial change to the traditional software development life cycle already adopted in an industrial setting. The related investment cost is often perceived as not justified in the light of the forecast benefits, and their adoption is not welcome both to managers, and to development teams more skilled in programming than in theorem proving.

Many efforts have been put by the B community in better integrating B in the traditional development process or with more popular methods, such as those based on UML, or in including advanced formal verification features. Just to give examples from the railway domain, we refer to (Metayer & Clabaut, 2008) and to (Marcano, Colin, & Mariano 2004). As applications of Event-B and UML-B, and to (Leuschel, Falampin, Fritz, & Plagge, 2009) for the integration of efficient property verification techniques derived by model-checking over B specifications.

Anyway, we can see that B-based methods have not so spread in railway software development, as one could have expected by their impressing record of successful applications in the domain.

On the other hand, accompanying the traditional life cycle with formal specification and verification techniques has often proved to have less impact and a better acceptance by managers and development teams pushed by the need to show compliance to CENELEC norms, although these techniques cannot in general promise the full formal proof of correctness achievable with a method that encompass the whole development, like B. We discuss in the following the techniques that have gained industrial acceptance in this respect, that is, Model Based Development, Model checking and Abstract Interpretation.

Figure 1. Stateflow model of a track circuit



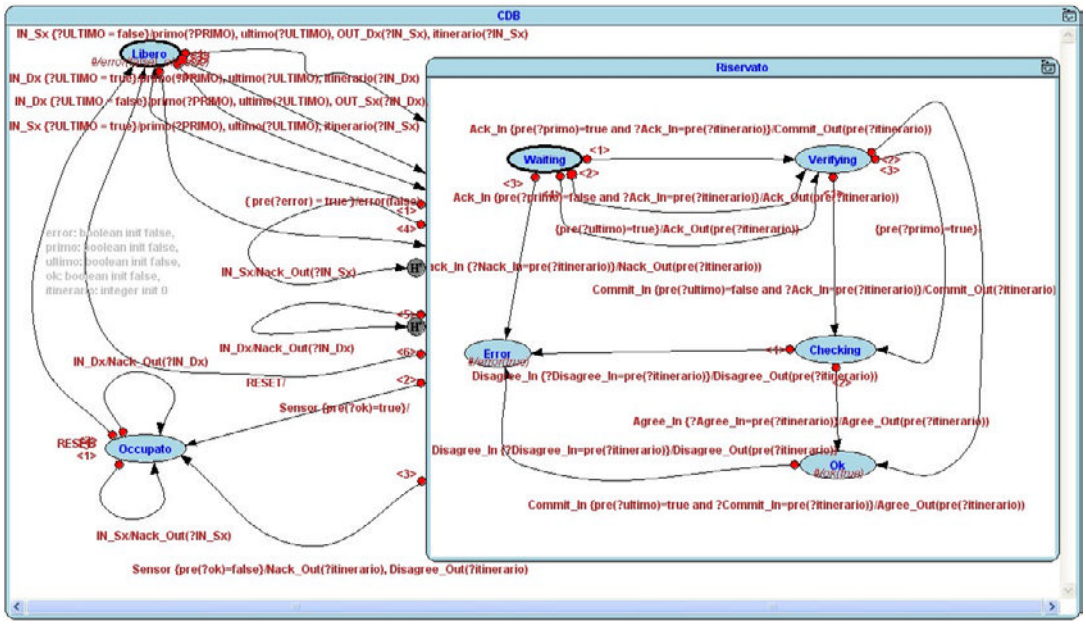
Model Based Development

The adoption of modelling technologies into the different phases of development of software products is constantly growing within industry. Designing model abstractions before getting into hand-crafted code helps highlighting concepts that can hardly be focused otherwise, enabling greater control over the system under development. This is particularly true in the case of embedded safety-critical applications industry which has been the first in line to adopt so called Model Based Development or Model Based Design (MBD), that employs modelling and simulation platforms like Simulink/Stateflow (toolboxes of Matlab from Mathworks) (SIMULINK) or the SCADE Suite (from Esterel Technologies) (Sauvage & Bouali, 2006) for lower-level design, to support the development of embedded applications. The adoption of automatic code generation, or automatic test cases generation, is also growingly followed in software production for safety critical systems.

A typical notation for modelling the discrete behaviour of a system is that of Statecharts, hierarchical extended finite state machines: introduced by Harel (Harel, 1987), they have specialized in various dialects, supported by formal specification environments: among the most adopted commercial environments we can find the mentioned Stateflow and SCADE. Figures 1 and 2 represent the model of the same system in the two latter tools: the diagrams are taken from students'exercises to model a track circuit of an interlocking system: such origin explains the use of Italian in some places, the differences in some modelling details, and the poor image quality, but clearly shows the similarity of state-based modelling. UML State Diagrams as well are essentially Statecharts, and are supported by several free and commercial tools.

One example from the railway signaling domain is the model based development cycle defined at the railway signaling division of General Electric Transportation Systems (GETS), inside a

Figure 2. SCADE model of a track circuit



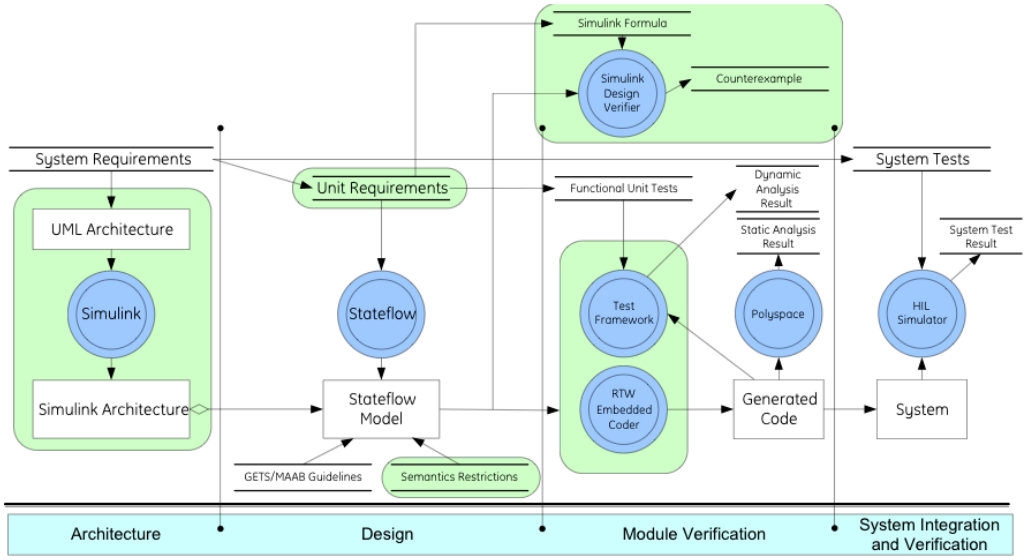
long-term collaboration with the University of Florence aimed at introducing formal methods to enforce product safety. The company employed modelling first for the development of prototypes (Bacherini, Fantechi, Tempestini, & Zingoni, 2006) and afterwards for requirements formalization and automatic code generation (Ferrari, Fantechi, Bacherini, & Zingoni, 2009). The production process for Automatic Train Protection (ATP) Systems is based on modelling by means of Simulink/Stateflow descriptions (see Figure 3).

A set of modelling guidelines has been introduced extending the MAAB guidelines (Matlab Automotive Advisory Board (MAAB), 2007), in order to restrict the Stateflow language to a semantically unambiguous subset. Extensive simulation of Stateflow diagrams with scenarios taken from the field is conducted, aiming at 100% structural coverage of the diagrams'states. After automatic code generation from the diagrams, back-to-back model/code testing is conducted by an automated tool, with the same simulation scenario. Back-to-back testing includes coverage comparison and has the main aim of confirming that the code generator has not introduced flaws in the code, being the code generator not validated against EN50128 (Ferrari, Magnani, Grasso, Fantechi, & Tempestini, 2011).

The kind of testing described above is one of the techniques that are encompassed by so called Model Based Testing (MBT). Another one is Automatic Test Generation (ATG) in which test cases are automatically generated from the model in order to guarantee an extensive coverage of the system functionalities as described by the model.

The SCADE suite as well has been widely adopted in the railway field: its usage is reported by Alstom, Ansaldo, Invensys, Siemens (SCADE). The activities that are supported by the suite are essentially the same as for Stateflow, but one point in favour for SCADE is that the suite includes a C Code Generator certified along the CENELEC EN50128. This allows in principle to eliminate from the development process those steps that in the previous example were aimed at guaranteeing safety of the code generator.

Figure 3. Simulink/Stateflow



Model Based Design is often not numbered among formal methods, essentially for two main reasons: the first is that some modelling frameworks are not based on a formal semantics, or they allow designers to write non precise models; the second reason is that no formal proof is given that the code is a correct concretization of the abstract specifications. This is not always the rule, and indeed models can be made precise by using semantically sound description formalisms (possibly resorting to modelling guidelines, as the mentioned MAAB for Stateflow). Formal verification of models can be conducted with the aid of model checking techniques, that we discuss in the next sections.

Model Checking

A formal verification technique that has recently acquired popularity also in industrial applications is Model Checking (Clarke, E.M., Grumberg, O. & Peled. D. 1999) an automated technique that, given a finite-state model of a system and a property stated in some appropriate logical formalism (such as temporal logic), checks the validity of this property on the model. Several temporal logics have been defined for expressing interesting properties. A temporal logic is an extension of the classical propositional logic in which the interpretation structure is made of a succession of states at different time instants. We consider here the popular CTL (Computation Tree Logic), a *branching time* temporal logic, whose interpretation structure (also called *Kripke structure*) is the *computation tree* that encodes all the computations departing from the initial states.

Formal verification by means of model checking consists in verifying that a Kripke structure M, modelling the behaviour of a system, satisfies a temporal logic formula ϕ, expressing a desired property for M. A first simple algorithm to implement model checking works by labelling each state of M with the subformulae of ϕ that hold in that state, starting with the ones having length 0, that is with atomic propositions, then to subformulae of length 1, where a logic operator is used to connect atomic propositions, then to subformulae of length 2, and so on. This algorithm requires a navigation of the state space, and can be designed to show a linear complexity with

respect to the number of states of *M.* One of the interesting features of model checking is that, when a formula is found not to be satisfied, the subformula labeling collected on the states can be used to provide a *counterexample,* that is, an execution path that leads to the violation of the property, thus helping the debugging of the model.

The simple model checking algorithm sketched above needs to explore the entire state space, incurring in the so called *exponential state space explosion*, since the state space often has a size exponential in the number of independent variables of the system.

Many techniques have been developed to attack this problem: among them, two approaches are the most prominent and most widely adopted. The first one is based on a symbolic encoding of the state space by means of boolean functions, compactly represented by Binary Decision Diagrams (BDD) (Bryant, 1986). The second approach considers only a part of the state space that is sufficient to verify the formula, and within this approach we can distinguish local model checking and bounded model checking: the latter has shown itself as particularly convenient since the problem of checking a formula over a finite depth computation tree can be encoded as a satisfiability problem, and hence efficiently solved by current very powerful *SAT-solvers* (Biere et al., 1999).

The availability of efficient model checking tools able to work on large state space sizes has favoured, from the second half of the nineties, their diffusion in industrial applications. The most known model checker of an academic origin are:

- **SMV:** Developed by the Carnegie Mellon University, for the CTL logic, based on a BDD representation of the state space (SMV);
- **NuSMV:** Developed by Fondazione Bruno Kessler, a re-engineered version of SMV which includes a Bounded Model Checking engine (NuSMV);
- **SPIN:** A model checker for the linear temporal logic LTL, developed at Bell Labs, for which the PROMELA language for the model definition has been designed (SPIN).

As we have already said, the revised EN50128 marks the first appearance of model checking as one of the recommended formal verification techniques in a norm regarding safety critical software.

In a later section we report about specific applications of model checking in the railway signaling domain, where this technique has been used to verify critical parts of the logic and of the design of such systems against safety related properties.

We wish to remember that model checking is a mere formal verification technique: sometimes is not considered a formal method, since it does not address the development of correct software, but only its verification. Model checking can be indeed adopted inside a software development process. In the application of model checking techniques in the development of software for safety critical applications, we can envisage two trends lines:

- As a verification activity on the models, inside a Model-Based development cycle, before deriving or generating from the models the final code;
- As a final verification on the produced code, whatever is the means for its production (Software Model Checking).

The first trend is mature for the industrial use, due to the inclusion of model checking engines inside commercial Model-Based development support tools, such as Simulink/Stateflow or SCADE. The second trend, which has seen till now few industrial applications, is actually a promise for the next future, thanks to several recent advances.

Model Checking Within Model Based Design

In a large part of the safety-critical systems industry, the Model Based Design approach has emerged as the main paradigm for the development of software. In this paradigm, early models are defined for the system and later refined to obtain detailed models from which code can be derived or automatically generated. Verification is conducted on the detailed models, either by extensive simulation (which ultimately corresponds to testing the code) with realistic simulation scenarios, which in some cases are pushed to interface the model with the hardware (the so called hardware-in-the-loop), or by formal verification.

Considering again the example of the GETS Model Based Development process, the company decided to perform a systematic experimentation with formal verification by means of Simulink Design Verifier, a test generation and property proving engine based on Prover Technology (Abdulla, 2004), that in its turn uses bounded model checking algorithms employing proprietary SAT-solvers. The desired property is actually expressed in the form of a graphical circuit where the variables observed by the property are input to Simulink blocks implementing logical, arithmetic and time delay operators. The verification engine checks that the formula is globally true for every execution path of the integrated Simulink/Stateflow model. If the property is violated, a counterexample showing a failing execution is given in the form of a test case for the model.

Similar experiences are reported with the use of SCADE; we recall in particular the one done at Airbus (Bochot et al., 2009), which has many points in common with the one of GETS, notwithstanding the different application domain. As in the case of Mathworks Stateflow Design Verifier, also Esterel Technologies SCADE Design Verifier is built on top of the proprietary very efficient SAT solver by Prover Technology. Also in this case, an observer based approach to property expression is adopted, this time using the SCADE synchronous logic blocks and gates.

An important experience of model checking within Model Based Design in the avionics domain is also the one by Rockwell Collins (Miller, Whalen, & Cofer, 2010), where both Simulink/Stateflow and SCADE were used as Model Based development environments, and several translators have been developed to apply different model checkers, among which SMV, NuSMV and, again, Prover, to models coming from different modelling tools. The first phases in this experience have confirmed that model checking is much more effective than testing in finding errors, and that BDD-based SMV and NuSMV are capable of dealing with very large state spaces. Challenging issues, like the treatment of models that include floating point arithmetic, have also been studied by defining suitable abstractions to deal with the complexity of floating point arithmetic and the implied huge size of the state space. Due to the resulting loss of precision, model checking once again has demonstrated itself very valuable for debugging, but not at demonstrating correctness.

We can say that the reported experiences show several convergences, such as the use of a few commercial MBD development environments, and related formal verification engines, which allow an expression of properties by means of the same (graphical) formalisms in which the models are designed, a feature particularly appreciated by designers. Also common conclusions are that verification is actually far from being a single push-button experiment. It is rather an iterative process, and it is far more likely able to detect errors in the model than to certify the absence of errors.

Software Model Checking

The first decade of model checking has seen its major applications in the hardware verification domain; meanwhile, applications to software have been made at system level, or at early software

design. Later, applications within the model-based development have instead considered models at a lower level of design, closer to implementation. But such an approach requires an established process hence excluding software written by hand directly from requirements: indeed, software is often received from third parties, who do not disclose their development process. In such cases direct verification of code correctness should be conducted, especially when the code is safety-related: testing is the usual choice, but testing cannot guarantee exhaustiveness.

Direct application of model checking to code is however still a challenge, because the correspondence between a piece of code and a finite state model on which temporal logic formulae can be proved is not immediate: in many cases software has, at least theoretically, an infinite number of states, or at best, the state space is just huge.

Pioneering work on direct application of model checking to code (also known as Software Model Checking) has been made at NASA since the late nineties by adopting in the time two strategies: first, by translating code into the input language of an existing model checker – in particular, translating into PROMELA, the input language for SPIN. Second, by developing ad hoc model checkers that directly deal with programs as input, such as JavaPathFinder (Visser et al., 2003). In both cases, there is the need to extract a finite state abstract model from the code, with the aim of cutting the state space size to a manageable size: the development of advanced abstraction techniques has only recently allowed a large scale application of software model checking. JavaPathFinder has been used to verify software deployed on space probes; in particular the detection and correction during the flight of a bug inside software on board of the Deep Space probe DS-1 has been reported (Havelund et al., 2000) Some software model checkers, such as CBMC (n.d.), hide the formality to the user by providing ”built-in” default properties to be proven: absence of division by zero, safe usage of pointers, safe array bounds, etc. On this ground, such tools are in competition with tools based on Abstract Interpretation, discussed in the next section.

It is likely that software model checking will in the next years gain a growing industrial acceptance also in the railway domain, due to its ability to prove the absence of typical software bugs, not only for proving safety properties, but also to guarantee correct behaviour of non safety related software.

Abstract Interpretation

Contrary to testing, that is a form of dynamic analysis focusing on checking functional and control flow properties of the code, static analysis aims at automatically verifying properties of the code without actually executing it. In the recent years, we have assisted to a raising spread of abstract interpretation, a particular static analysis technique. Abstract interpretation is based on the theoretical framework developed by Patrick and Radhia Cusot in the seventies (Cousot & Cousot, 1977). However, due to the absence of effective analyses techniques and to the lack of sufficient computer power, only after twenty years software tools have been developed to support it so that applications of the technology at industrial level could take place. The focus of the application of the technology is mainly on the analysis of source code for runtime error detection, which means detecting variables overflow/underflow, division by zero, dereferencing of non-initialized pointers, out-of-bound array access and all those errors that, might them occur, bring to undefined behaviour of the program.

Since the correctness of the source is not decidable at the program level, the tools implementing abstract interpretation work on a conservative and sound approximation of the variable values in terms of intervals, and consider the state space of the program at this level of abstraction. The problem boils down to solve a system of equations that represent an over-approximate version

of the program state space. Finding errors at this higher level of abstraction does not imply that the bug also holds in the real program. The presence of false positives after the analysis is actually the drawback of abstract interpretation that hampers the possibility of fully automating the process. Uncertain failure states (i.e., statements for which the tool cannot decide whether there will be an error or not) have normally to be checked manually and several approaches have been put into practice to automatically reduce these false alarms. The process developed at GETS, illustrated in Fig.3, includes abstract interpretation analysis, employing the Polyspace tool (Deutsch, 2004). False positives are handled through a step of abstraction refinement able to filter out most of them (Ferrari, Magnani, Grasso, Fantechi, & Tempestini, 2011).

RAILWAY SIGNALING SOFTWARE

We have already noticed that the recommendations from CENELEC EN 50128 guidelines are graduated along the SIL of the developed software. Also formal methods are strongly recommended only for higher SIL components. In railway applications, the higher SIL is usually assigned to railway signaling equipments and components. We will follow this indication, by concentrating on this kind of equipment, that has seen a number of successful applications of the techniques described above, and provides the most challenging opportunities. Indeed, scalability issues often hamper the industrial application of formal methods, and in particular formal verification; many successful examples of verification through model checking do not scale easily to cope with the large size of modern computer applications. In order to discuss how this issue has manifested itself in the domain of railway signaling, we first follow a traditional distinction between two large classes of applications in this domain:

- Train control systems, that guarantee safe speed and braking control for trains (like ATP and ATO[1])
- Interlocking systems, that establish safe routes through the intricate layout of tracks and points of a railway station.

Several other minor safety-related signaling systems could be considered, which are often used to provide input for main signaling systems are for this reason customized for the particular devices and do not exhibit scalability issues. Axle counters may be classified in this category when they are simple train presence sensors, but axle counter system may include train control functionalities as well, and in this case they should be classified in the first category above.

Monitoring systems, diagnosis systems and supervision systems (such as ATS – Automatic Train Supervision) may have a large complexity, but are usually not assigned a high or medium SIL. This does not mean that these systems would not benefit by the introduction of formal methods: it is the opinion of the writer that indeed a wider use of formal specification and verification definitely improves the quality of such software, but in many cases at the price of higher costs. It is where the very high costs of extensive testing alone are forced by safety certification, that formal methods do promise to achieve better safety objectives, with less cost.

In the end, we also admit that some signaling systems actually merge features of both categories above, and this seems to be the next future trend, which we will address later. For the purpose of our next discussion, it will however be useful to keep these two major classes separate.

Train Control Systems

A variety of train control systems exist, with different degree of authority over the driver, different means to convey information to the train and different nature of this information. Typically, an

ATP/ATO train speed control system consists of wayside devices, which transmit a telegram that contains the data to be processed by the car-borne equipment, and of the car-borne equipment, which receives the telegram data and performs the actual enforcement of train speed. The most notable example of this category of systems is ERTMS/ETCS (ETRMS), but several systems are in operation, such as the ATP systems for secondary lines, that can be considered as cut down cases of ETCS, and specific solutions are often developed for metros.

However, the basic general principle on which train speed control is based is common: the braking curve concept. The position of the preceding train on a line, or of a fixed obstacle, defines a curve for the maximal safe speed of the train in any point of the line at a given time. The train has to maintain its own speed below the curve. A violation of this rule should end in applying brakes. This safety principle, implemented within the on-board equipment, is consolidated and easily formalized.

The main challenge for safety is instead to make sure that knowledge on board of the train regarding the curve is sufficiently accurate. The focus is therefore on the safety and real-time performance of wayside - on board transmission. Proper interface definition and formal communication protocol specifications are needed. We can at this regard cite the systematic approach to the formalization of ETCS natural language requirements that has been recently attempted by the Eu-RailCheck project by the European Railway Agency (Cavada et al. 2009).

In the end, in this class of systems scalability of formal verification is naturally achieved by the separation of concerns typical of the nature of the systems, and by proper definition of communication interfaces and protocols (Esposito et al. 2003).

Interlocking Systems

A different story can be told for interlocking systems. An interlocking is the safety-critical system that controls the movement of trains in a station and between adjacent stations. The interlocking monitors the status of the objects in the railway yard (e.g., points, switches, track circuits) and allows or denies the routing of trains in accordance with the railway safety and operational regulations.

In order to facilitate the representation of the interlocking logic by signaling engineers, often the traditional and well-established relay-based principle diagrams are referred as the only trusted source of information for computer-based interlocking developers, and conformance of new interlocking systems to such sources is addressed by means of costly and tedious, but possibly not exhaustive, testing. Following this trend, one of the most common ways to define the interlocking logic is through boolean equations (or, equivalently) ladder diagrams, which are interpreted either directly by a PLC or by a proper evaluation engine over a standard processor. A first concern in computerized interlocking has been the generation of such Boolean equation sets, starting from the generic signaling principles and from the topology of the layout of the station (Fringuelli et al. 1992). But we can say that anyway the specification of the logics of an interlocking system is by nature formal.

In principle, model checking can be applied to such equations systems to verify that they do not violate safety requirements; as an example, one of the typical safety properties that is normally required to be verified of a railway interlocking system is the *no-derailing* property: "while a train is crossing a point, the point shall not change its position".

However, due to the high number of boolean variables involved, automatic verification typically incurs in combinatorial state space explosion problems. The problem is due to the fact that interlocking rules are strongly dependent among them, making centralized resolution of an enormous system of boolean equations the only choice. The first applications of model checking have therefore attacked portions of an interlocking

systems (Bernardeschi et al., 1998)(Cimatti et al., 1998)(Gnesi et al., 2000)(Groote et al., 1995). But even recent works (Winter & Robinson, 2003) (Fantechi et al. 2010) show that routine verification of interlocking design for large stations is still out of reach of standard verification tools, although several advances with dedicated tools (such as the ones based on SAT decision procedures) or with domain-dependent optimized strategies (Winter 2012) have been achieved,

Indeed Prover Technology (Prover) has launched a commercial solution for the production of interlocking software, that includes formal proof, by means of a SAT solving engine, of safety conditions. Figures about the addressable size of the controlled yard are not yet known to the writer.

Geographic approaches to the definition of the logic of an interlocking, such as EURIS (Berger, Middelraad & Smith, 1993), or the proposal in (Banci & Fantechi, 2005) allow for a less monolithic and distributed definition, on which formal verification techniques can arguably be better exploited.

THE FUTURE ROLE OF FORMAL METHODS

Technological advances push in every field towards a collection of smaller, but interconnected systems: a well know example is that of Wireless Sensor Networks. On the contrary, interlocking systems have been traditionally built as centralized systems that contain the whole logic driving all the physical entities that are deployed on a quite large area. In some recent applications, a large station is divided in regions, each controlled by a separate interlocking system, still of fairly large dimensions; or, some local logic is delegated to some field controllers that are in charge each of a small area but the central interlocking is responsible of the global safety logic. In both cases, as we have shown, the size of the logic challenges the current verification capabilities.

The current, and next future, trend towards distributed solution is likely to affect interlocking solution as well, pushing the use of small controllers, distributed over the field, linked by some kind of (wireless?) safe connection. The interlocking logic itself would be distributed. Mentioned geographic approaches to the specification of interlocking functions would help in this case. But safety assessment of the global logic would require at least: the demonstration of safety of the overall global logic principles, the (easy) proof by model checking of the (small) controllers, and the proof of safety of the communication protocols. In such a more and more distributed scenario it is likely that what was once upon a time a monolithic interlocking system developed with proprietary methods will be based on a network of controllers possibly coming from several providers, that need to interoperate: another reason, safety apart, to adopt formal specification of interfaces and formal verification of conformance of implementations.

On the other hand, integration of such local controllers with ATP/ATO systems will also be a challenge, and will need to be attacked basing on well defined interfaces. The currently running FP7 INESS project, (INESS 2009) aims at defining the specifications for a new generation of interlocking systems, with particular attention to properly interface ERTMS systems: in this case UML State Diagrams have been chosen as the modelling language.

Another challenge pushed by the technological trends will probably be the coexistence on the same (possibly distributed) systems of application with radically different SIL. An immediate example is a train network on which both vital and not vital information is transferred. To avoid the costs of validation of the whole system at the highest SIL, trusted separation mechanism will be needed: formal specification and verification of such mechanisms will allow to achieve such trust.

CONCLUSION

We have presented a gallery of snapshots on the wide activity about the industrial application of formal methods in the railway domain. We have shown that some trends in formal specification and verification have consolidated and are ready for industrial usage. In particular, we have noticed two main streams, the more consolidated application of B-based methods, and the emerging formal verification activities inside Model-based processes. However the evolution of the domain, the evolution of software tools, the evolution of regulations, either for interoperability and for safety, the European integration, and the opening of the markets are posing new challenges. In particular, it is still difficult to identify a standard formal method, or a standardized development process based on formal methods; domain dependent and application dependent methods will likely need still to be used.

We have not discussed for space reasons a number of issues related to formal methods adoption. We want to remark for example the important issue of the long term life of software development and verification tools that is needed to support maintenance through the entire life (measured in decades, not years) expected for railway equipment: relying on a proprietary tool suite is often perceived as a danger for the long-term maintenance, thus suggesting to continue to use old time traditional and simple software handcrafting tools.

ACKNOWLEDGMENT

This chapter is based in part on work done inside a collaboration with General Electric Transportation Systems. In particular I wish to thank A. Ferrari, D. Grasso and G. Magnani for their assistance on the parts of this chapter devoted to the experience in GETS. I also thank the students C. Borgiotti, G. Calamai, D. D'Amico, N. Giannerini for their help with statecharts snapshots.

REFERENCES

Abdulla, P. A., Deneux, J., Stalmarck, G., Agren, H., & Akerlund, O. (2004). Designing safe, reliable systems using SCADE. *LNCS 4313: IsoLA 2004,* (pp. 115–129), Paphos, Cyprus. Berlin, Germany: Springer.

Abrial, J. R. (1996). *The B-Book*. Cambridge University Press. doi:10.1017/CBO9780511624162

Bacherini, S., Fantechi, A., Tempestini, M., & Zingoni, N. (2006). A Story about Formal Methods Adoption by a Railway Signaling Manufacturer. In J. Misra, T. Nipkow, & E. Sekerinski (Eds.), *LNCS 4025: FM 2006: Formal Methods* (pp. 179-189), Hamilton, Canada. Berlin, Germany: Springer.

Banci, M., & Fantechi, A. (2005). Geographical vs. functional modelling by statecharts of interlocking systems. *Electronic Notes in Computer Science*, *133*, 3–19. doi:10.1016/j.entcs.2004.08.055

Behm, P., Benoit, P., Faivre, A., & Meynadier, J. M. (1999). Météor: A successful application of B in a large project. *LNCS 1708: World Congress on Formal Methods in the Development of Computing Systems* (pp. 369–387). Toulouse, France. Berlin, Germany: Springer.

Berger, J., Middelraad, P., & Smith, A. J. (1993). EURIS, European railway interlocking specification. [Institution of Railway Signal Engineers.]. *Proceedings IRSE*, *93*, 70–82.

Bernardeschi, C., Fantechi, A., Gnesi, S., Larosa, S., Mongardi, G., & Romano, D. (1998). A formal verification environment for railway signaling system design. *Formal Methods in System Design*, *12*(2), 139–161. doi:10.1023/A:1008645826258

Biere, A., Cimatti, A., Clarke, E. M., & Zhu, Y. (1999) Symbolic model checking without BDDs. *LNCS 1579: Tools and Algorithms for Construction and Analysis of Systems*, (pp. 193-207). Amsterdam, The Netherlands. Berlin, Germany: Springer.

Bochot, T., Virelizier, P., Waeselynck, H., & Wiels, V. (2009). Model checking flight control systems: The Airbus experience. *ICSE Companion*, *2009*, 18–27.

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, *C-35*(8), 677–691. doi:10.1109/TC.1986.1676819

Cavada, R., Cimatti, A., Mariotti, A., Mattarei, C., Micheli, A., Mover, S., et al. Susi, A. & Tonetta S. (2009) EuRailCheck: Tool Support for Requirements Validation. *ASE 2009* (pp. 665-667), Auckland, New Zealand. Washington D.C.: IEEE Computer Society

CBMC. (n.d.), Retrieved from: http://www.cprover.org/cbmc/

CENELEC. (2001). *EN 50128, Railway Applications - Communications*. Signaling and Processing Systems - Software for Railway Control and Protection Systems.

Cimatti, A., Giunchiglia, F., Mongardi, G., Romano, D., Torielli, F., & Traverso, P. (1998). Formal verification of a railway interlocking system using model checking. *Formal Aspects of Computing*, *10*(4), 361–380. doi:10.1007/s001650050022

Clarke, E.M., Grumberg, O. & Peled. D. (1999) *Model Checking*. MIT PRESS.

Cousot, P., & Cousot, R. (1977). Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages* (pp.238-353), Los Angeles, CA. New York, NY: ACM Press.

DaSilva, C., Dehbonei, Y., & Mejia, F. (1993) Formal specification in the development of industrial applications: Subway speed control system. *5th IFIP Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE'92)*, (pp. 199–213). Perros-Guirec, France. Amsterdam, The Netherlands: North-Holland.

Deutsch, A. (2004) Static verification of dynamic properties. Polyspace white paper.

ERTMS. (n.d.) Retrieved from: http://www.ertms.com.

Esposito, R., Lazzaro, A., Marmo, P., & Sanseviero, A. (2003) Formal verification of ERTMS Euroradio safety critical protocol. *4th Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'03),* Budapest. L'Harmattan Hongrie.

Fantechi, A., Fokkink, W., & Morzenti, A. (2011). Some Trends in Formal Methods Applications to Railway Signaling. In Gnesi, S., & Margaria, T. (Eds.), *Formal Methods for Industrial Critical Systems: A Survey of Applications*. Wiley-IEEE Computer Society Press.

Ferrari, A., Fantechi, A., Tempestini, M., & Zingoni, N. (2009). Modelling Guidelines for Code Generation in the Railway Signaling Context. *Proceedings of 1st NASA Formal Methods Symphosium (NFM)* (pp 166-170). Moffet Field, CA, USA.

Ferrari, A., Grasso, D., Magnani, G., Fantechi, A., & Tempestini, M. (2010). The Metro Rio ATP case study. S. Kowalewski, & M. Roveri (Eds.), *LNCS 6371: 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS 2010)*, (pp. 1-16)Antwerp, Belgium. Berlin, Germany: Springer.

Ferrari, A., Magnani, G., Grasso, D., & Fantechi, A. (2010) Model checking interlocking control tables, *Proc. 8th FORMS/FORMAT symposium* (pp. 98-107)

Ferrari, A., Magnani, G., Grasso, D., Fantechi, A., & Tempestini, M. (2011). Adoption of Model-Based Testing and Abstract Interpretation by a Railway Signaling Manufacturer. *International Journal of Embedded and Real-Time Communication Systems*, *2*(2), 42–61. doi:10.4018/jertcs.2011040103

Fringuelli, B., Lamma, E., Mello, P., & Santocchia, G. (1992). Knowledge-based technology for controlling railway stations. *IEEE Intelligent Systems*, *7*(6), 45–52.

Groote, J. F., Koorn, J. W. C., & van Vlijmen, S. F. M. (1995) The safety guaranteeing system at station Hoorn-Kersenboogerd. *10th IEEE Conference on Computer Assurance (COMPASS'95),* (pp. 131-150). Washington D.C.: IEEE Computer Society, Gnesi, S., Latella, D., Lenzini, G., Abbaneo, C., Amendola, A. & Marmo, P. (2000) An automatic SPIN validation of a safety critical railway control system. *IEEE International Conference on Dependable Systems & Networks,* (pp. 119-124). Washington D.C.: IEEE Computer Society.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231–274. doi:10.1016/0167-6423(87)90035-9

Havelund, K., Lowry, M., Park, S. J., Pecheur, C., Penix, J., Visser, W., & White, J. L. (2000) Formal Analysis of the Remote Agent Before and After Flight. *5th NASA Langley Formal Methods Workshop,* Williamsburg, Virginia. INESS project (2009), Retrieved from: http://projects.uic.asso.fr/JavaPathFinder. http://javapathfinder.sourceforge.net/

Lecomte, T. (2008). *LNCS 5014: FM 2008* (pp. 430-434), Turku, Finland. Berlin, Germany: Springer.

Leuschel, M., Falampin, J., Fritz, F., & Plagge, D. (2009) Automated Property Verification for Large Scale B Models. *LNCS 5850: FM 2009* (pp. 708-723). Eindhoven, The Netherlands. Berlin, Germany: Springer.

MAAB. (2007). *Control Algorithm Modelling Guidelines Using Matlab, Simulink and Stateflow*, Version 2.0. Retrieved August 7, 2009, from http://www.mathworks.com/industries/auto/maab.html

Marcano, R., Colin, S., & Mariano, G. (2004). A Formal Framework for UML Modelling with Timed Constraints: Application to Railway Control Systems. *LNCS 3297: SVERTS: Specification and Validation of UML models for Real Time and Embedded Systems*, (pp.33-38) Lisbonne, Portugal. Berlin, Germany: Springer.

Metayer, C., & Clabaut, M. (2008). DIR 41 Case Study. *LNCS 5238: Abstract State Machines, B and Z, First International Conference,* (p. 357), London, UK. Berlin, Germany: Springer.

Miller, S. P., Whalen, M. W., & Cofer, D. (2010). Software model checking takes off. *Communications of the ACM*, *53*(2), 58–64. doi:10.1145/1646353.1646372

NuSMV. (n.d.), Retrieved from: http://nusmv.fbk.eu

Prover Technologies. (n.d.), Retrieved from: http://www.prover.com/

Sauvage, S., & Bouali, A. (2006) Development Approaches in Software Development. *Proceedings of ERTS*, Toulouse, France.

SCADE. (n.d.), Retrieved from: http://www.esterel-technologies.com/

SIMULINK. (n.d.), Retrieved from: http://www.mathworks.com/products/simulink/

SMV. (n.d.), Retrieved from: http://www.cs.cmu.edu/~modelcheck/smv.html

SPIN. (n.d.), Retrieved from: http://spinroot.com/spin/whatispin.html

Visser, W., Havelund, K., Brat, G., Park, S. J., & Lerda, F. (2003). Model Checking Programs. *Automated Software Engineering*, *10*(2), 203–232. doi:10.1023/A:1022920129859

Winter, K., (2012) *Symbolic Model Checking for Interlocking Systems, Railway Safety, Reliability and Security: Technologies and Systems Engineering*,

Winter, K., & Robinson, N. J. (2003) Modelling Large Railway Interlockings and Model Checking Small Ones. *Proceedings of the 26th Australasian Computer Science Conference* (pp. 309–316).

ENDNOTES

1. ATP = Automatic Train Protection, protects the train from dangerous misbehaviours of the driver;
2. ATO = Automatic Train Operation, driverless operation of the train.

Chapter 13
Symbolic Model Checking for Interlocking Systems

Kirsten Winter
The University of Queensland, Australia

ABSTRACT

Model checking is a fully automated technique for the analysis of a model of a system. Due to its degree of automation it is in principle suitable for application in industry but at the same time its scalability is limited. Symbolic model checking is one approach that improves scalability through the use of Binary Decision Diagrams (BDDs) as an internal data structure. This approach allows the user to increase the efficiency by customising the ordering of state variables occurring in the model to be checked. In the domain of railway interlockings represented as control tables, it is found that this task can be supported using an algorithm that has access to the track layout information. In our work we propose optimisation strategies that render symbolic model checking feasible for large scale interlocking systems.

Our results yield a verification tool suitable for use in industry.

INTRODUCTION

Railway signalling interlockings are safety critical systems. They are designed to permit the safe movement of trains along a railway system. Therefore special attention has to be given to the correctness of the design and the implementation of an interlocking system. In order to guarantee a safe functioning of the system railway engineers are currently manually validating the Interlocking against the Signalling Principles.

DOI: 10.4018/978-1-4666-1643-1.ch013

The development of such systems is very labour intensive and prone to error. It requires specialised skills. Moreover, possible errors in the design are detected very late in the design process. To mitigate these problems Queensland Rail (QR), the major railway operator and owner in Queensland, Australia, intended to support its design process by a specialised tool set called the Signalling Design Toolset (SDT) (which was first introduced in Robinson, Barney, Kearney, Nikandros & Tombs (2001)). Parts of this toolset were intended for supporting the verification task.

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Railway interlockings, next to hardware designs, have been shown earlier to be a suitable application for automated verification techniques, in particular model checking and automated theorem proving (Groote, Koorn & van Vlijmen, 1995; Eisner, 1999; Boralv & Stålmarck, 1999; Huber & King, 2002; Simpson, Woodcock & Davies, 1997). In our work we propose to integrate model checking into the design process to ensure the correctness of a design before it is implemented.

Model checking is a fully automated technique for the analysis of a model of a system. Due to its degree of automation it is suitable for application in industry but at the same time its scalability is limited. For complex system models checking the complete state space often leads to the state explosion problem, a situation where run-time and memory usage of the checking process exceed the tolerable limits (as too many states are to be explored). This problem has been targeted by researchers in the last twenty years. The aim is to improve the efficiency of the algorithms and push the boundary of feasible models that can be checked automatically.

Symbolic model checking (McMillan, 1993; Burch, Clarke, McMillan, Dill & Hwang, 1992) is one of the early proposals for improving the efficiency of model checking. In this approach the model, its states and transitions between states (i.e. its behaviour) are represented by a data structure called Binary Decision Diagrams, BDDs (Bryant, 1986). This data structure has the advantage that in some cases its representation can be reduced to a structure of very small size. Consequently, less memory space is necessary and the operations to compute and check the state space become very fast. What is needed, however, is a good ordering of the variables that describe the model to allow for a sufficient - if not optimal - reduction of the BDDs.

Queensland Rail (QR) contracted the Software Verification Research Centre in 2002 to conduct a feasibility study on the automated analysis of control tables. Control tables instantiate the generic signalling and operational principles for interlocking systems in a particular country or region (Queensland Rail Signal and Operational Systems, 1998). Each instantiation is given with respect to a particular geographical layout of the railway equipment. We performed this research in close cooperation with the railway engineers. The results of the feasibility study (Winter, 2002a) were in favour of the approach of symbolic model checking and using the tool NuSMV (Cimatti, Clarke, Giunchiglia, Giunchiglia, Pistore, Roveri, Sebastiani & Tacchella, 2002). They support the claim that symbolic model checking scales sufficiently well if enhanced by a good user-defined variable ordering. For control tables a very good ordering can be generated from the layout information that is available. An algorithm that can process this layout information can generate the ordering and therefore, once set up, the process is fully automatic. The improvement of efficiency is for some models more than ten-fold.

Others have applied model checking to analyse railway interlocking systems: Gnesi, Lenzini, Latella, Abbaneo, Amendola & Marmo (2000), Bernardeschi, Fantechi, Gnesi & Mongardi (1996), and Cleaveland, Luettgen & Natarajan (1996), for instance, have analysed fault tolerance of interlocking systems. The main focus in their work was communication aspects between components rather than the control logic as in our work.

The first approaches on applying model checking to verify railway interlocking systems represented as control tables or equations were reported by Groote, Koorn & van Vlijmen (1995) using μCRL and its tools, and Eisner (1999; 2002) using SMV. Also Simpson, Woodcock & Davies (1997), and Huber & King (2002) are focusing on the control logic similar to our approach. In particular, the approach by Eisner (1999; 2002) and Huber & King (2002) is very close to ours as they also use a symbolic model checker. The interlocking systems, however, are modelled on a lower level of abstraction. This leads to significantly different models and in particular a more

complex model of the requirements which in our case can be simply expressed as train collision and train derailment.

Ferrari, Magnani, Grasso & Fantechi (2011) attempt a comparison of different model checkers when applied to interlocking control tables. Although the work is very interesting it is lacking to take into account the potential of optimising symbolic model checking as it has been proposed in this work.

In this chapter we will describe the process of model checking control tables symbolically. The work is based on our experience when using the model checker NuSMV to check control tables developed at QR. We will give a detailed recipe for generating a very good (if not optimal) variable ordering that dramatically outperforms the ordering generated by the NuSMV tool based on heuristics. Furthermore, we propose an extension of the NuSMV model checker that allows the user also to improve the ordering of the transitions. Beyond that our experiments lead to further insights of how to adjust the standard user options of the NuSMV tool for best results in memory usage and computation time. The results that are presented in this chapter consolidate work published in Winter & Robinson (2003), Winter, Johnston, Robinson, Strooper & van den Berg (2005) and Johnston, Winter, van den Berg, Strooper & Robinson (2006).

The structure of the chapter is organised as follows. Section "Modelling of Control Tables and Signalling Safety Principles" specifies the scope of our work and introduces control tables as a target of our analysis. In Section "Modelling of Control Tables and Signalling Safety Principles" we introduce the concept of BDDs and variable orderings and transition clustering. We show how to generate a variable ordering for BDDs that leads to a very efficient encoding of control table and track layout information and how to organise the transition clusters. We report the results of resources used by the model checker when applying the techniques discussed to three different sized models. Finally, the last section concludes the work and gives an outlook on future work.

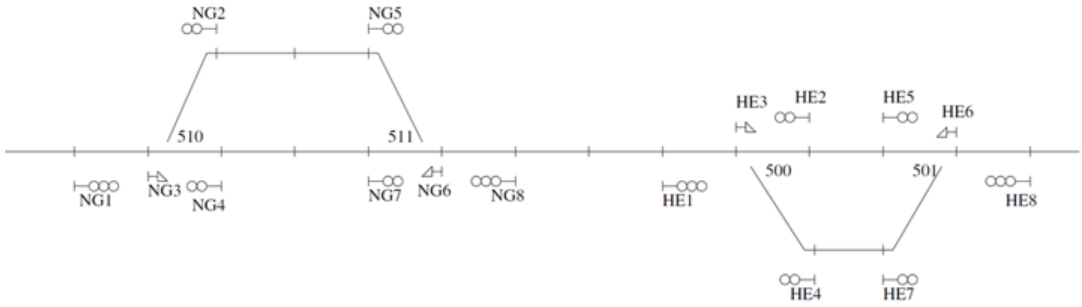
MODELLING OF CONTROL TABLES AND SIGNALLING SAFETY PRINCIPLES

Railway signalling permits trains to move safely through a railway network. The network consists of tracks and other signalling equipment like signals and points. How tracks, signals and points within a certain area are located with respect to each other is captured in the track layout of that area. A track layout abstracts from the actual length of the tracks and distances between the signalling equipment in the layout. Figure 1 shows an example containing the signals, e.g., HE1 and HE2, and points, e.g., 510 and 511, and tracks which are not named in the figure but indicated as sections on the railway.

A railway interlocking prescribes the behaviour of the signalling equipment in a specific area which in our context is called the verification area and monitors the position of trains on the layout via track circuits. The functionality of QR's interlockings is generally specified by QR's signalling principles (Queensland Rail Signal and Operational Systems, 1998). The functionality of an interlocking for a particular verification area is specified by a set of control tables. These tables specify the control for the signals, points and routes of a particular track layout and can be read as a collection of rules under which the equipment can be safely operated. Figure 2 shows a part of the (signal) control table for the layout shown in Figure 1. It specifies the conditions for the route from signal HE2 to signal NG8.

Control tables provide the abstract specification of railway interlockings and serve as a basis for the system development. Other approaches to verifying the logic of interlocking systems target the interlocking software itself or the geographi-

Figure 1. Track Layout of a small verification area



cal data model (e.g., Boralv & Stålmarck (1999), Eisner (1999), and Huber & King (2002)). For the sake of an early detection of errors during the development of an interlocking, and the benefit of having a simpler, more abstract model, we target the high-level specification and check the control table and the entries in its columns instead.

To achieve this we initially created a generic model of the (signal) control table using the formal modelling notation ASM (Gurevich, 1995). This control table model has been extensively discussed and validated by QR's railway engineers. This was possible due to the nature of the ASM notation which suited the problem well and allowed us to readily express the interlocking functionality of the control table format: The model consists of a collection of conditional rules which can be understood by railway engineers. The model, after being consolidated and validated by the domain experts, provides a formal semantics of QR's control table notation. As shown in Winter (2008), ASM models translate into SMV code (McMillan, 1993) which is then input to the model checker NuSMV (Cavda, et al., 2010).

The ASM model consists of a static part and a behavioural part. The static part collects information on signals, points, tracks etc. from the

Figure 2. Control Table for the small verification area as in Figure 1 (shown is the row for route he2_1m with its sub-rows)

Signal	Route Number	Route to	Route Indication	Requires									Replaced by Tracks Occ
				Points Locked		Routes		Route Holding	or Until		Tracks		
				Normal	Reverse	Normal	Reverse	Maintained by Tracks occ	Tracks occ	for Time secs	Clear	Occ	
he2	1m	ng8	-	p500							HE1BT HE1AT HE2AT HE2BT HE2CT NG7BT NG7AT NG8AT		HE1BT
						he3(1s)		HE1BT					
						he1(1M) he1(2M)		HE1BT HE1AT					
						ng5(1M) ng7(1M)		HE1BT HE1AT HE2AT HE2BT HE2CT NG7BT NG7AT					

Figure 3. Transition Rule for Route Reverse

```
transition Route_Reverse ==
   do forall r in Routes
      if
         guard(r)
      then
         routelock(r) := rtR
      endif
   enddo
```

track layout and the control table of a particular verification area.

Commands from the control centre are modelled as input variables that change non-deterministically. That is, in any situation, there can be a command to change the status of a particular point or signal (Winter, Johnston, Robinson, Strooper & van den Berg, 2005). The behavioural part models the logic of interlocking control as it is described through the control table. This part is generic in that it is the same for every verification area. It comprises a set of conditional rules which are fired synchronously. The conditions of each rule prescribe under which circumstances (parts of) the equipment under control can change its status. Figure 3 shows the structure of these rules using the rule for setting routes reverse as an example: For all routes *r*, if a particular condition, *guard*(*r*), is satisfied then we will set the routelock of the route to reverse (*rtR*). The condition *guard*(*r*) is a conjunction of simple conditions on *r* that can be extracted from the control table. Other rules for e.g., setting routes normal or changing the aspect of a signal are modelled similarly.

The framework of this model is generic. To create a model for a particular control table we instantiate the static part of the model with the data of a particular verification area. In the later stages of the project (after consolidation of the ASM model) we shortened the process and worked on the SMV code level. That is, we have a generic SMV model (initially created from the ASM model) that is to be instantiated with the data for the targeted verification area. The instantiation process has been automated within the SDT tool suite (Robinson, N., Barney, D., Kearney, P., Nikandros, G. & Tombs, D., 2001).

Unlike other approaches our model also includes (one or two) trains moving along the tracks. As a consequence, the safety requirements become generic and very easy to validate: they are modelled in terms of trains colliding or derailing. It could be shown through tests that even a very simplistic model of train movement suffices to show missing entries in the control table. For example, we do not model the speed of a train or its braking capacity and that it might overrun red signals. In our model trains move according to the conditions of the points and signals from track to track and can stop at any time. The train data is limited to an identifier, which route the train is on, and which track it is occupying. It suffices to consider only two trains in the system to check for collision, and only one train to detect possible derailment. The reasoning for this simplification, which is in agreement with the railway engineers,

is based on the fact that the more trains are running through a particular verification area the more the movement of other trains is restricted. Moreover, trains can appear on the tracks of the verification area in an arbitrary fashion in our model. Therefore, every possible combination of two potentially colliding trains is investigated. Considering only two trains at maximum limits the additional complexity that stems from adding trains to the model to a tolerable level. For more details on our model the reader is referred to Winter & Robinson (2003) and Winter (2002b).

SYMBOLIC MODEL CHECKING WITH CUSTOMISED ORDERINGS

Symbolic model checking was first introduced by Burch, Clarke, McMillan, Dill and Hwang (1990) and the proposed algorithms later implemented by McMillan in the SMV tool (McMillan, 1993). The tool NuSMV (Cimatti, et al., 2002) is an open-source development that is based on the SMV tool and extends its initial features.

Central to symbolic model checking is the idea of symbolically representing the model (i.e., its states and state transitions) using Bryant's Binary Decision Diagrams (BDDs) (Bryant, 1986). BDDs are a graph structure that allows the canonical representation of Boolean functions. In most cases a BDD representation is substantially more compact than other canonical representations in normal form (e.g., conjunctive and disjunctive normal form).

States and state transitions of a system can be modelled as Boolean relations using a Boolean encoding of the (state) variables and their values. We denote the set of Boolean variables with V. A state is described by the evaluation of variables in V. A Boolean function over V can be identified with the set of evaluations of the variables that make the function true. Hence, a Boolean function over V is suitable for representing a set of states and can be thought of as a BDD. In the following we call this Boolean function S, i.e., $S(V)$ denotes a set of states.

To describe the transitions of the system from one state to a next state we use a second set of variables V' which contains the next state variables (i.e., the state variables after the transition has taken place). In a similar fashion as above, an evaluation of variable pairs in V and V' describes the transition relation. We denote the transition relation as $N(V, V')$.

Binary Decision Diagrams

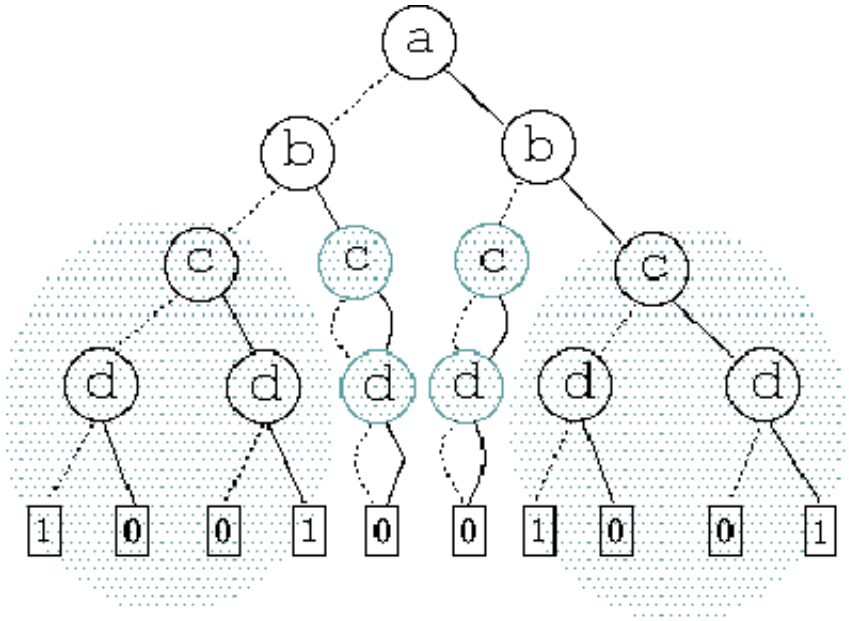
A BDD consists of a set of nodes that are labelled with a variable. Each node is linked via two directed edges to its successor nodes. These edges represent the evaluation of the variable to false and true, respectively. The leaf nodes of the BDD are labelled with 0 and 1, representing the truth values false and true. If the labels of the nodes in the BDD are ordered then we call the graph an *ordered* BDD (OBDD).

Figure 4 shows an example of an OBDD that represents the function $(a \leftrightarrow b) \wedge (c \leftrightarrow d)$. The variables encountered when traversing the graph from the root to the leaves are strictly ordered using the ordering

$a < b < c < d$. The dotted edges in the figure indicate the evaluation of a node variable to false and the full edges to the right indicate the evaluation to true. The leaf nodes indicate the evaluation of the paths that lead to them.

The graph in Figure 4 can be reduced as it contains redundant information. For instance, if both edges of a node lead to the same successor (e.g., the shaded nodes in the tree) then the evaluation of this node is obviously irrelevant for the value of the function along this path. Also if sub-trees in the structure are identical then they need to be represented only once (see shaded sub-trees in the figure). The reduced graph is called a *reduced* OBDD (ROBDD) and is shown in Figure 5 on the left hand side. The reduction is done automatically. The resulting ROBDD is significantly smaller and

Figure 4. Complete OBDD that represents the function (a ↔ b) ^ (c ↔ d)



is again a canonical representation which allows us to represent states and transitions.

The possible reduction of an OBB depends on its structure which is determined by the ordering of its node variables. An OBDD for the same function but with the changed variable ordering $a < c < b < d$ has a different shape and consequently reduces differently. The resulting ROBDD for this ordering is shown in Figure 5 on the right hand side. Clearly, the reduction is not as effective and the ROBDD is substantially bigger than the ROBDD for the original ordering.

Optimising Variable Orderings

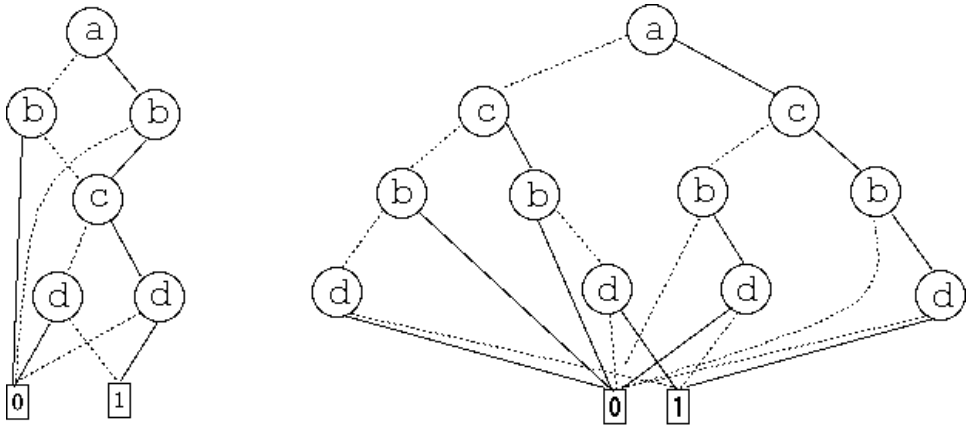
It can be shown that it is infeasible to compute a variable ordering that is optimal for the reduction of an OBDD. However, much research effort has been focused on heuristics for finding a good variable ordering automatically.

The general ideas behind good orderings are based on dependencies between variables and early evaluation of the represented formula. When a formula represents a transition that updates one variable it will reference variables on which the transition depends. We call these variables *support variables* of the transition. Variables that are referenced by all transitions are called global variables.

The following guidelines summarise what often makes a good ordering of variables. (Note that in NuSMV code all possible updates of a variable are usually handled by one transition, i.e., each variable has a corresponding transition.)

- Declare closely related variables together. In the variable ordering, each variable should be close to the support variable of its transition [Clarke, Grumberg, Peled, 2000; Lewis, Comella-Dorda, Gluch, Hudak & Weinstock, 2001).
- For each transition, having the support variables closer to the top of the order than the variable being transformed, gives the smallest ROBDD (Clarke, Grumberg, Peled, 2000).

Figure 5. Reduced OBDDs that represent the function (a ↔ b) ^ (c ↔ d): Left hand side with ordering a < b < c < d and right hand side with ordering a < c < b < d.



- Place global variables at the top of the order (Lewis, Comella-Dorda, Gluch, Hudak & Weinstock, 2001).
- Initially order variables manually and run the automated re-ordering procedures on-the-fly (i.e., during model checking) whenever the size of the graphs exceed a certain threshold. This functionality is called dynamic reordering (Kamhi and L. Fix, 1998) and is provided by NuSMV.

Some of the guidelines require input from the domain expert whereas others refer to techniques that are often automated within the model checker. Recent trends in the research of tool development focus mostly on automated techniques. However, input from the domain expert is still possible and can be exploited if available. In the domain of railway interlockings this is the case and will lead to best results as we will see later.

Ordering of Transition Relations

Due to the nature of the application our model has similarities with the domain of synchronous hardware circuits: The transition relation is formed by a set of single transitions each describing the update of one state variable if certain conditions (which depend on other state variables) are satisfied. That is, the value for a state variable *v* can be given as

$$v = f(v_1 \ldots, v_n)$$

where *f* is a function that captures the logical dependencies between variable *v* and other state variables $v_1, \ldots, v_n$.

All single transitions are fired simultaneously which leads to a next state in which some variables have changed their values while others not. Therefore, we can use results that originated in the domain of synchronous circuit verification, in particular the results by Burch, Clarke & Long (1991).

In symbolic model checking the state space is explored by iteratively applying the transition relation which is the conjunct of all single transitions to the state. This is done in a forward fashion starting with the initial state and is called image computation. The operation on ROBDDs used for image computation is called *relational product*. For synchronous systems (as in our case) it is defined as follows.

$$S'(v'_1, \ldots, v'_n) = \exists v_1 \epsilon V \ [\ldots \exists v_n \epsilon V \ [S(v_1, \ldots, v_n) \Box N(v_1, \ldots, v_n, v_1', \ldots, v_n') \ ] \ \ldots]$$

where $v_1, \ldots, v_n$ is the set of primed state variables and $S'(v_1', \ldots, v_n')$ describes the set of next states reachable from S($v_1, \ldots, v_n$) via one transition step. We abbreviate this formula using the notation from

Burch, Clarke & Long (1991) as follows

$$S'(V') = \exists_{v \in V} [S(V) \Box N(V, V')] \quad (1)$$

where V' is the set of primed state variables and $S'(V')$ describes the set of next states reachable from $S(V)$ via one transition step.

The transition relation $N(V, V')$ can be applied to the state as one big transition or it can be envisaged as a conjunct of smaller partitions of N, $N_i(V, V')$, where each partition is either a single transition (that updates one variable) or a conjunct of several single transitions, referred to as cluster in the following. In practice each $N_i(V, V')$ can often be represented by a small BDD whereas the whole of $N(V, V')$ becomes very large. The aim is to compute the image without building the whole of $N(V, V')$ by using the following formula:

$$S'(V') = \exists_{v \in V} [S(V) \Box N_0(V, V') \Box \ldots \Box N_{n-1}(V, V')] \quad (2)$$

Unfortunately, existential quantification (e.g., $\exists_{v \in V}$ in the formula above) does not distribute over conjunction and we cannot simply split up the operation into single steps. Burch et al. in (Burch, Clarke & Long, 1991), however, developed a method to overcome this problem. It is based on two observations. Firstly, our model of interlockings exhibits locality (in a similar fashion to circuits), that is, most single transitions will depend on only a small number of variables in V and V'. Secondly, sub-formulas from the relational product can be moved out of the scope of the quantification if they do not depend on the variables being quantified. Therefore, it is beneficial to conjoin the $N_i(V, V')$ with $S(V)$ one at a time moving out those variables from the scope of the quantification that none of the remaining partitions depends on. To do so we want to order the partitions $N_i(V, V')$ in such a way that the number of variables that can be eliminated early is maximised. This will lead to smaller intermediate results of the image computation.

Assume the chosen order of clusters is given by the permutation p that permutes the indices $\{0, \ldots, n\text{-}1\}$.

That is, cluster $\boldsymbol{N}_{p_i}(V, V')$ will be applied in the ith step of the image computation. Let Dp(i) be the set of variables that $\boldsymbol{N}_{p_i}(V, V')$ depends on and E_i the set of variables that can be eliminated after the ith step. E_i can be computed as follows:

$$E_i = D_{p(i)} - \sum_{k=i+1}^{n-1} D_{p(k)} \quad (3)$$

Then $S'(V')$ can be computed in a number of steps each eliminating the corresponding variables E_i and building an *intermediate product* $S_{i+1}(V, V')$*:*

$$\begin{aligned} S_1(V, V') &= \exists_{v \epsilon E_0} [S(V) \Box N_{p(0)}(V, V')] \\ S_2(V, V') &= \exists_{v \epsilon E_1} [S_1(V) \Box N_{p(1)}(V, V')] \\ S'(V') &= \exists_{v \epsilon E_{n-1}} [S_{n-1}(V, V') \Box N_{p(n-1)}(V, V')] \end{aligned} \quad (4)$$

The chosen order p has an impact on how early variables can be quantified out and therewith affects the size of the BDDs constructed. The aim is to group those transitions together into one cluster that have the same support variables. Selective grouping of transitions into clusters, and the order p of application of the clusters leads

to smaller and fewer intermediate products that are manipulated faster (Geist & Beer, 1994). The following heuristics can be proposed: Transitions that are supported by the maximal number of variables should be grouped together in a cluster and applied first. Subsequent transitions that are supported by fewer variables should be grouped into clusters so that as many of their support variables as possible do not support transitions in clusters yet to be applied. This enables some of the support variables to be quantified out progressively from the intermediate products giving smaller intermediate products.

If transitions do not naturally fall into clear-cut divisions, the grouping of transitions within clusters and the order of application of the clusters should be such that early elimination of support variables is maximised.

We will see in the next section how a good ordering for the transitions can be generated for the domain of railway interlockings.

Customising the Orderings for Railway Interlockings

In our initial attempts of checking control tables we found that applying the model checker to larger models quickly lead to either a memory overflow or an unacceptable run-time. In some cases the checking process did not terminate at all. We had to improve on the efficiency of the process in order to be useful for checking QR's railway interlockings.

We addressed this aim by customising the variable ordering and the order of transition partitions and their clustering. The following sections describe how this was successfully done in our project.

Characteristics of the Data

Variables in our model can be divided into three groups: global, local and input variables. Similarly, transitions can be characterised as global or local.

Variables

Global variables represent train attributes like the current position (given in terms of a track) and the currently used route. In the SMV code this is modelled by four variables of enumerated type. Typically

30 -130 different values, depending on the number of tracks and routes in the interlocking, are required.

The larger and more complex the verification area, the larger becomes the set of Booleans necessary to represent the values. Typically five to seven Booleans are required for each attribute in the implementation (see Clarke, Grumberg & Peled (2000) for details on implementing enumerated types efficiently).

Local variables model the lie of the individual points, the current aspect of the signals, and the lock and usage of routes. Mostly, this information can be represented by simple Booleans (e.g., points are set normal or reverse, signals are set *proceed* or *stop*, routes are locked normal or reverse), only the route usage is encoded by a typically small enumerated set.

Input variables represent signalling and train control commands (i.e., requests). They are not controlled by the interlocking but change their values randomly (to capture every possible behaviour). This can be modelled using a number of simple Booleans variables and one variable of enumerated type. The number of enumerated values again depends on the size of the verification area, i.e., the number of points, signals and routes. The implementation of the enumerated input variable typically requires five to seven Booleans and can be thus considered a large variable.

An increase in the complexity of models (more signals, points and tracks), introduces more local variables, and maintains the same number of global and input variables but adds more values to the enumerated types. Adding more values to the enumerated types does not impact significantly on the number of Booleans used to implement them

but does impact on the size of the ROBDD used to distinguish particular values of the variables.

Transitions

The transition relation in our model is described using the next operator of the SMV input language (McMillan, 1993; Cavda, Cimatti, Jochim, Keighren, Olivetti, Pistore, Roveri & Tchaltsev, 2010). For each variable the evaluation in the next state is modelled depending on the previous values of a number of support variables. The size of the ROBDD representing the transition for each variable is dependent on the variable ordering (see Section "Optimising Variable Orderings").

Transitions for global variables are called global transitions. They are supported by all the variables. Transitions for local variables, local transitions, depend on a limited number of variables. Specifically they are supported by the global variables, the input variables and some of the other local variables. For example, only the occupation of particular nearby or local tracks and the input command variable are relevant to the movement of a particular point.

An analysis of the dependencies between all the variables using a dependency matrix (Moon, Hachtel & Somenzi, 2000) resulted in a very dense matrix.

The Variable Ordering

Let $Var = \{v_1, \ldots, v_{m+n+p}\}$ be the set of state variables in our model with $\{g1, \ldots, gm\} \subset$ Var the set of global variables, $\{l1, \ldots, ln\} \subset$ Var the set of local variables and $\{req1, \ldots, reqp\} \subset$ Var the set of input variables. Let N_{v_i} (V, vi'), $1 \leq i \leq (m + n)$, be the transitions, local or global, that changes (local or global) variable vi dependent on support variables $V \subseteq Var$.

In our project the transitions are such that if $\{v_1, v_2, v_3\}$ is the set of support variables for transition N_{v_1} (V, v1') then the set of support variables for N_{v_3} (V, v3') is likely to include v1. That is, there is a cross-dependency between transitions and it is not obvious which variable should come first. However, local transitions which model the update of signals, points, etc. as shown in the track-layout will depend on the support of signalling equipment that is in the close vicinity on the track-layout graph. The dependencies between the state variables are therefore related to the *geographical* arrangement that can be read from the track layout (see, for example, Figure 1).

To define an ordering that reflects the dependencies between the state variables the layout is viewed as a grid and the signals and points are read in order from left to right. Where signals or points are in the same vertical grid, elements are ordered from top to bottom.

Mechanical interlocking design suggests a tailoring of this ordering strategy that is based on the layout. The order of building the relays for the mechanical interlockings was to start with points, followed by the signals whose routes crossed those points. As typically each signal is associated with routes from that signal and so these routes are listed along with the signal, this strategy also associates the signals and routes with a particular point. The associated entities form a group.

It was noted that moving one signal and its routes from a group with its first in-route point, which was a trailing point, to a group with its first in-route facing point made a significant difference to the time taken and the memory used. However, in some interlockings there are several different ways of associating signals and points. The nearest in-route point geographically for a signal may have the routes from the signal cross it in a trailing direction or a facing direction. There may be several in-route points for a route. These considerations led to the following first ordering heuristics: A signal and its routes are associated with the first facing point in the route, and where

there is only a trailing point in-route, with that point. The groups around the points were kept in geographic order.

While variables within groups are related by the transition relations, there is significant cross dependencies between the groups e.g. routes are related to routes that oppose them and the opposing routes are likely to be associated with different points. This leads us to the second ordering heuristics: The groups are best ordered according to their arrangement on the track layout rather than in a random order.

These two heuristics form the basis of our ordering strategy, called geographic order. Ordering the local variables {l1, …, ln} ⊂ Var according to the geographic order, defines a permutation ϒ for the local variables. This leads to an ordering of local variables of the form lϒ(1) < … < lϒ(n). For each local variable $l_{\gamma(j)}$, 1 ≤ j ≤ n, the corresponding transition $N_{l_{\Upsilon(j)}}$ (V, l'ϒ(j)) then depends on local variables in reasonably close proximity to lϒ(j) in the order, e.g., $l_{\Upsilon(j-1)}$ and $l_{\Upsilon(j+1)}$, etc.

The local variables also depend on the global variables. Experimentation shows that putting the global variables higher in the variable order than all the local variables gives the smallest local transitions (supporting heuristics 3 in Section "Optimising Variable Orderings"). The transitions for the four global variables of enumerated type depend on all the variables and are large.

Placement of the input variables in the variable order is problematic. Input variables are in the support variables for all transitions. When they are placed at the beginning of the order, the ROBDD representing the transitions N_{v_i} (V, vi'), 1 ≤ i ≤ (m + n), are smaller than ROBDDs for an order in which the input variables are placed lower in the order. However, this does not necessarily lead to smaller intermediate products. Experimentation has shown that placing the large input variable (see Section "Characteristics of the Data") lower in the order increases the size of the local transitions and the size of the clusters. However, this gives smaller intermediate products and uses less memory overall. There are time and memory efficiency penalties for manipulating large transitions, large clusters, and large intermediate products and for our data, experimentation has shown that the best results are obtained by placing the large input variable about 2/3 down the order. For a detailed discussion see Johnston, Winter, van den Berg, Strooper & Robinson (2006). We have implemented an algorithm that performs this ordering using the available data from the track layout.

The Transition Ordering and Clustering

NuSMV did not have provision for the user to supply a transition order at the time we started our project. It has its own generic algorithm for estimating the affinity of transitions (Moon, Hachtel & Somenzi, 2000) (which describes their degree of similarity) and by default progressively builds clusters based on this affinity. A cluster is closed off when its size reaches a threshold that the user supplies or the default threshold. This results in evenly sized clusters.

For railway interlockings the dependency matrix on which the affinity is based is very dense. The behaviour of all variables is heavily interrelated. Therefore, computing the affinity between variables by itself did not provide the necessary information to improve efficiency. However, examining the railway interlocking model and its semantics has enabled us to define an order in which transitions can be conjoined and the points in the order at which to cut the conjunctions to form clusters. When these clusters are applied in turn in the image computation, the variables

are quantified efficiently from the intermediate product.

Transition Ordering

In our model of the control table, the global transitions, N_{g_1}, …, N_{g_m}, are supported by all the other variables including the input variables. The reasoning in Section "Ordering of Transition Relations" suggests that global transitions should be applied first. The local transitions, N_{l_1}, …, N_{l_n}, depend on global variables and other local variables associated with nearby symbols in the track layout, an argument that was used to define the geographic variable ordering (see Section "The Variable Ordering"). A transition order that reflects the geographic order of variables for the local transitions results in a permutation $N_{l_{\Upsilon(1)}} < \ldots < N_{l_{\Upsilon(n)}}$ of local transitions which then can be progressively grouped into clusters with some overlap of support variables. That is, the same argument of vicinity of symbols on the track layout that is used for finding a good variable ordering can be reused for ordering the partitioned transition relation.

Eliminating variables that are at the leaf end of an ROBDD (lowest in the variable order) favours BDD reduction and results in smaller diagrams than removing variables from the middle or root end of the diagram (higher in the variable order). Therefore, we order the local transition in such a way that transitions for variables of lower order will be applied first. If the local variables indexed progressively by $\Upsilon(1)$, …, $\Upsilon(n)$ using the geographic order Υ then the aim is that the transition for the $\Upsilon(n)$th variable is applied before the transition for the $(\Upsilon(n\text{-}1))$th variable to facilitate early elimination of the $\Upsilon(n)$th variable. While the $\Upsilon(n)$th variable may not be eliminated immediately after application of its transition, it should be soon after since all transitions using it will be within close range. This leads to an ordering of transitions that is similar to the variable ordering but the transitions are applied in reverse order for the local variables.

Generally, a good order of application of transitions is the global transitions followed by the local transitions in the order $\Upsilon(n)$ to $\Upsilon(1)$. That is, assuming the NuSMV principle of prepending the cluster list and applying the transitions from the back to the front of the list, a good transition order for railway interlockings is the local transitions in the order $\Upsilon(1)$ to $\Upsilon(n)$, followed by the global variable transitions:

$$N_{l_{\Upsilon(1)}} < \ldots < N_{l_{\Upsilon(n)}} < N_{g_1} < \ldots < N_{g_m} \qquad (5)$$

The NuSMV code was extended so that the user could provide a transition order in terms of an ordered list of the corresponding variables, $l_{\Upsilon(1)} < \ldots < l_{\Upsilon(n)} < \ldots < g_1 < \ldots < g_m$.

Forming the Clusters

Transitions are conjoined in order according to the transition order. Having defined a good transition order that supports the elimination of variables as early as possible, the question becomes where to cut the transition conjunction and form a cluster. If all transitions are in one cluster, no elimination of variables can occur and the ROBDD representing the cluster becomes very large. If the clusters are too small then many intermediate products $S'(V')$ (see Equation 4 in Section "Ordering of Transition Relations") have to be computed. The issue is to find the balance between size and number of clusters and intermediate products.

Using a transition order and the default threshold to form the clusters resulted in between ten and fourteen clusters for our models. This number is too large and we had to re-define the cut-off points for the clusters.

The global transitions are applied first and it is logical to put all of these into the first cluster. After

Table 1. Comparison of various sized models using the discussed options

	User Options	Time(secs)	Memory Used
Small model	1	4081	65Mb
	2	651	95Mb
	3	124	42Mb
	4	61	29Mb
	5	88	36Mb
Medium Model	1	9620	1098Mb
	2	734	114Mb
	3	321	78Mb
	4	152	49Mb
	5	222	63Mb
Large model	1	N/A	ran out of memory
	2	N/A	ran out of memory
	3	68872	3.6Gb
	4	33641	980Mb
	5	29357	1160Mb

application of the global cluster the next values for the global variables can be quantified out.

When clustering the remaining local transitions, which are ordered using the geographical ordering, we used the insight that by referencing the track layout it is possible to nominate where in the transition order the dependencies change. This observation was confirmed by the railway engineers. For example,

Figure 1 shows us that variables related to symbols to the right of signal HE1 will be supported mostly by variables lower in the transition order than the variable for signal HE1 since we ordered the variables inspecting the track layout from left to right. Similarly, variables related to symbols to the left of signal HE1 will be mostly supported by variables higher in the variable ordering than the variable for signal HE1. Thus, for this verification area the local transitions fall naturally into two clusters at this point. Including the global cluster gives three clusters for this track layout.

The code for NuSMV was extended to allow the user to define clusters. The suggested approach reduced the run-time by a half and reduced the memory usage by a third for a medium sized model compared to previous results.

In general we found the models fell naturally into three or four clusters. However, for large models these clusters can become very big and the model checker spent significant time building them. In this case the performance was best using a clustering based on the threshold.

From our experimentation it is clear that with a good transition order, few clusters are required. Another way to achieve few clusters is to specify a large threshold. This approach is not as efficient as the customised formation of clusters described above (as the clusters will not be cut as precisely as before) but is a worthwhile improvement on the default threshold used by standard NuSMV. The result (shown in Table 1 in Section "Experimental Results") suggests that this approach is a reasonable alternative as it requires no specialist knowledge of the model or the application domain.

Experimental Results

Table 1 compares our results for three different sized models: The *large model* consists of 41 routes, 9 points, 19 signals, and 31 track circuits. The *medium model* comprises 29 routes, 9 points, 13 signals, and 22 tracks. The *small model* comprises 12 routes, 2 points, 8 signals, and 8 tracks.

The experiments were conducted using the following options:

Option 1: Using NuSMV defaults for variable and transition orders and clustering
Option 2: Using user-defined variable order with default transition order and clustering
Option 3: Using user-defined variable ordering and user-defined transition orders with default clustering
Option 4: Using user-defined variable ordering, user-defined transition order and clusters selected by user
Option 5: U sing user-defined variable order, user-defined transition order and clusters selected by threshold.

The figures show that a large improvement over run-time and memory usage was achieved by choosing a good variable ordering that was based on geographical information from the track layout, i.e., domain knowledge over the dependencies. This result is not surprising as this correlation is often stated in the literature.

Improvements of similar scale could also be achieved by customising the order of transition partitions and by forming the clusters. Both parameters were chosen using the same reasoning as was used for choosing the variable ordering - in our case geographic order of dependencies.

The NuSMV tool (from version 2.4.1) has been extended by the user option **-t <tv_file>** which allows the user to specify an alternative variable ordering to be used for clustering of the transition relation (Cavda, et al., 2010). The grammar of the ordering file is the same as for the variable ordering file.

CONCLUSION AND FUTURE WORK

This chapter proposes a method for improving the efficiency of symbolic model checking for railway interlocking systems. It is based on the idea to include the user's domain knowledge for providing a good ordering for variables and transitions.

The domain knowledge necessary can be easily extracted by tracing the geography of the layout of the targeted verification area. This can be done automatically. The findings are based on a careful analysis of the data and the process. The experimental results provide evidence for the potential of symbolic model checking for the analysis of railway interlockings.

For future work it would be interesting to include an optimised symbolic model checking approach into a comparative study as it has been done in Ferrari, Magnani, Grasso & Fantechi (2011). Moreover, we would like to apply the proposed strategies to a model on the level of code or a geographical data model of an interlocking system to confirm whether similar improvements as for the control table model can be achieved.

ACKNOWLEDGMENT

The results presented in this chapter are based on a series of joint projects with Queensland Rail. The author would like to thank George Nikandros, David Barney and David Tombs from Queensland Rail for their interest in our work and many fruitful discussions. The results would not have been possible without Neil Robinson's patience when translating railway terminology to me and most of all the great persistence of Wendy Johnston who did not get tired of looking at BDDs and drawing up very large dependence matrices.

I would also like to thank the anonymous reviewers for their valuable comments and their effort in reading the manuscript thoroughly.

REFERENCES

Bernardeschi, C., Fantechi, A., Gnesi, S., & Mongardi, G. (1996). Proving safety properties for embeddedcontrol systems. In *Procs. of Conference on Dependable Computing* (EDCC-2), *16*(440), 321-332. Springer-Verlag.

Bryant, R. E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE TransactionsOn Computers*, C-35(8), IEEE. Boralv, A. & Stålmarck, G. (1999). Formal verification in railways. In Hinchey, M., & Bowen, J. (Eds.), *Industrial-Strength Formal Methods in Practice*. Springer-Verlag.

Burch, J., Clarke, E., & Long, D. (1991). Symbolic model checking with partitioned transition relations. In *Int. Conf. on Very Large Scale Integration*.

Burch, J. R., Clarke, E., McMillan, K., Dill, D., & Hwang, L. (1992). Symbolic model checking1020 states and beyond. *Information and Computation*- Special issue: Selections from 1990IEEE symposium on logic in computer science, *98*(2), 142-170.

Cavda, R., Cimatti, A., Jochim, C. A., Keighren, G., Olivetti, E., Pistore, M., et al. (2110). *NuSMV 2.5 User Manual*. Retrieved March, 7, 2011 from http://nusmv.irst.itc.it.

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). NuSMV 2: an OpenSource tool for symbolic model checking. In Brinksma, E. & Larsen, K. G., (Eds.), *Proc. of Int. Conference on Computer Aided Verification* (CAV 2002),*volume 2404 of LNCS*, (pp. 359-364) Springer Verlag.

Clarke, E., Grumberg, O., & Peled, D. (2000). *Model Checking*. MIT Press.

Cleaveland, R., Luettgen, G., & Natarajan, V. (1996). Modeling and verifying distributed systems using priorities: A case study. In *Procs. of Int. Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96), volume 1055 of LNCS,(* pp. 287-297). Springer-Verlag.

Eisner, C. (1999). Using symbolic model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. In *Proc. of Conf. on Correct Hardware Design and Verification Methods (CHARME'99), volume 1703 of LNCS.* Springer-Verlag.

Eisner, C. (2002). Using symbolic CTL model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. *Software Tools for Technology Transfer*, *4*(1), 107–124. doi:10.1007/s100090100063

Ferrari, A., Magnani, G., Grasso, D., & Fantechi, A. (2011). Model checking interlocking control tables. In Schnieder, E. & Tarnai, G. (Eds.), *Proceedings of Conference on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 2010), 2*, (pp. 107 – 115). Springer-Verlag.

Geist, D., & Beer, I. (1994). Efficient model checking by automated ordering of transition relation. In Dill, D., (Ed.), *Proc. of Int. Conference on Computer-Aided Verification* (CAV'94), *volume 818 of LNCS*, (pp. 299-310.) Springer-Verlag.

Gnesi, S., Lenzini, G., Latella, D., Abbaneo, C., Amendola, A., & Marmo, P. (2000). An automatic SPIN validation of a safety critical railway control system. In *Procs. of IEEE Conference on Dependable Systems and Networks*, (pp. 119-124.) IEEE Computer Society Press.

Groote, J. F., Koorn, J. W. C., & van Vlijmen, S. F. M. (1995). The safety guaranteeing system at station Hoorn-Kersenboogerd. In *Proceedings 10th IEEE Conference on Computer Assurance* (COMPASS95), (pp. 131-150). IEEE Computer Society Press.

Gurevich, Y. (1995). Evolving Algebras 1993: Lipari Guide. In Börger, E. (Ed.), *Specification and Validation Methods*. Oxford University Press.

Huber, M., & King, S. (2002). Towards an integrated model checker for railway signalling data. InEriksson, L.-H. & Lindsay, P. (Eds.), *Proc. on Formal Methods Europe* (FME'2002), *volume2391*, (pp. 204-223.) Springer-Verlag.

Johnston, W., Winter, K., van den Berg, L., Strooper, P. A., & Robinson, P. (2006). Model-based variable and transition orderings for efficient symbolic model checking. In Misra, J., Nipkow, T. & Sekerinski, E. (Eds.), *Proc. of 14th Int. Symposium on Formal Methods* (FM 2006), *volume 4085 of LNCS*,(pp. 524-540.) Springer-Verlag.

Kamhi, G., & Fix, L. (1998). Adaptive variable reordering for symbolic model checking. In *Proc. of IEEE/ACM Int. Conference on Computer-aided design* (ICCAD'98), (pp. 359-365.) ACM Press.

Lewis, G., Comella-Dorda, S., Gluch, D., Hudak, J. & Weinstock, C. (2001). *Model-based verification: Analysis guidelines. Technical Report CMU/SEI-2001-TN-028*, Carnegie Mellon Software Engineering Institute.

McMillan, K. (1993). *Symbolic Model Checking*. Kluwer Academic Publishers. doi:10.1007/978-1-4615-3190-6

Moon, I., Hachtel, G. D., & Somenzi, F. (2000). Border-block triangular form and conjunction schedule in image computation. *In Proc. of Formal Methods in Computer-Aided Design* (FMCAD 2000), *volume 1954 of LNCS*,(pp. 73-90), Springer-Verlag.

Queensland Rail Signal and Operational Systems. (1998). *Signalling Principles - Brisbane SuburbanArea. Technical Report S0414,* Queensland Rail Technical Services Group.

Robinson, N., Barney, D., Kearney, P., Nikandros, G., & Tombs, D. (2001). Automatic generation and verification of design specification. In *Proc. of Int. Symp. of the International Council On Systems Engineering* (INCOSE).

Simpson, A., Woodcock, J., & Davies, J. (1997). The mechanical verification of solid state interlocking geographic data. In Groves, L. & Reeves, S. (Eds.), *Proc. of Formal Methods Pacific* (FMP'97), Discrete Mathematics and Theoretical Computer Science Series, pp. 223-243. Springer-Verlag.

Winter, K. (2002). *Feasibility study on control table verification. SigTools-041, version 0.4*, October 2002.

Winter, K. (2002b). *Model checking control tables: the ASM-NuSMV approach. SigTools.039, version 0.1*, October 2002.

Winter, K. (2008). *Model Checking Abstract State Machines*. VDM Verlag.

Winter, K., Johnston, W., Robinson, P., Strooper, P., & van den Berg, L. (2005). Tool support for checking railway interlocking designs. In Cant, T. (Ed.), *Proc. of the 10th Australian Workshop on Safety Related Programmable Systems (SCS'05), volume 55*, (pp. 101-107). Australian Computer Society, Inc.

Winter, K., & Robinson, N. J. (2003). Modelling large railway interlockings and model checkingsmall ones. In Oudshoorn, M. (Ed.), *Proc. of Australasian Computer Science Conference* (ACSC2003).

KEY TERMS AND DEFINITIONS

Binary Decision Diagrams: Data structure for concise representation of data and its efficient manipulation.

Control Table: Abstract specification of an interlocking.

Model Checking: Automated analysis technique that fully explores the state space of a model.

NuSMV: A symbolic model checker.

Railway Interlocking: Control system to permit the safe movement of trains along a railway system.

Safety Critical System: System whose failure can lead to harm, or loss of life or property.

Symbolic Model Checking: Model checking approach that is based on the use of Binary Decision Diagrams.

Variable Ordering: Order of variables in a Binary Decision Diagram which determines its shape and size.

Section 6

Human Factors

Chapter 14
Designing Usable Interactive Systems within the Railway Domain:
A Human Factors Approach

Nina Jellentrup
German Aerospace Center, Institute of Transportation Systems, Germany

Michael Meyer zu Hörste
German Aerospace Center, Institute of Transportation Systems, Germany

ABSTRACT

Train drivers as well as signallers interact with several computer based information and communication systems to ensure safe and effective train operations. So far the technical progress mostly determines the design of such interactive systems and requirements out of a human factors perspective are not integrated. Beside the development of technical functions it is essential to take the usability as a quality attribute of every interactive system into account. If the usability is not considered during system development, it could occur that there are several functions available within a system but the user does not know how to use them in an efficient way. This chapter describes a psychological approach to design or redesign usable interactive systems within the railway domain. Some examples will be discussed to demonstrate the approach and the results.

INTRODUCTION AND BACKROUND

In the beginning of the railways more than 175 years ago operation and use of the technology was limited by the technical possibilities at that time. The operation was done manually and safety systems were more or less inexistent. Increasing speeds and complexity of the operation heightened the workload on the operators. A long history of invention and improvement of safety systems led from that time to today. Nowadays many technical systems support the driver as well as the signal-

DOI: 10.4018/978-1-4666-1643-1.ch014

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

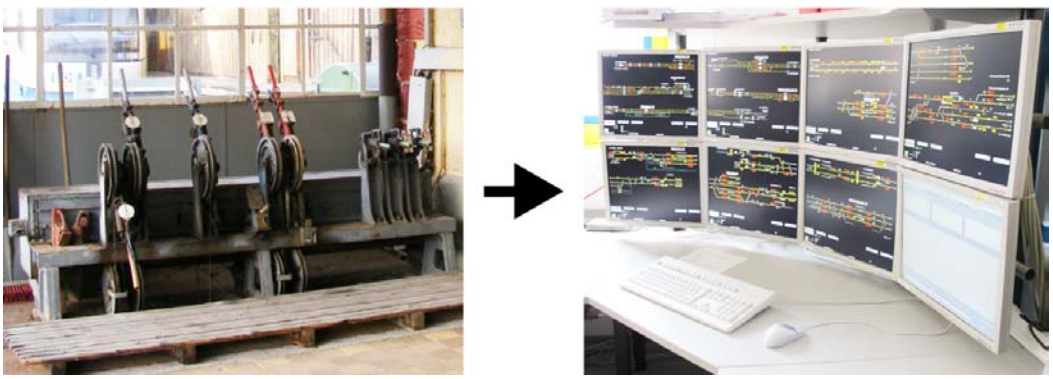
ler in their work. Nevertheless human operators play still an important part of the railway system.

The railway system is traditionally characterized by technical developments and innovations but the human factors influencing the performance of the operators have long been paid little attention. Through continuous automation it was often tempted to reduce the influence of human actions more and more - but not all tasks can be replaced or hedged by technical systems. Therefore human contribution is still an essential part of the railway system. In recent years the technical progress changed the work of many railway employees. While switches and signals were previously set by the local signaller with mechanical levers, today the route setting is done with the computer-based electronic interlocking workstation (Figure 1).

Similarly, the train driver does not enter the train data no longer by means of mechanical switches, but using an interactive system. Today the interaction with different interactive systems defines the work of many employees in the railway domain. Train drivers as well as signallers interact with several computer based information and communication systems to ensure safe and effective train operations. The user interface of such an interactive system can be defined as the language through which the system and the user interact with each other. This usually concerns the design of the display and the feedback provided by the system (system to user language). Furthermore the users indicate to the system what they want to do via input devices such as a mouse or a keyboard (user to system language). So far mostly the technical progress determined the design of such user interfaces within the railway domain and requirements from a human factors perspective are barely considered. However, it is not always sufficient simply to ensure that essential information is given and that specific functions are available within a system. Beside the development of technical functions it is essential to take the usability of the user interface as a quality attribute of every interactive system into account. Although usability is often understood as 'user-friendly' it does not actually imply that a system 'is friendly to the user'. Usability encompasses much more: it is a quality attribute of a user interface that is more or less present in every user interface. This includes the user interfaces of the railway domain as well. Therefore usability aspects should be considered in the design of interactive systems within the railway domain as well. The usability engineering approach - which will be described in more detail within this chapter - provides structured methods for the integration of usability aspects within system development.

ISO 13407 defines that Usability is 'the extent to which a system can be used by a specific

Figure 1. Changes in the signalling workplace



user group in a specific context of use to achieve specific tasks in an effective, efficient and satisfactory manner' (ISO 13407, 1999). Whereas in this context effectiveness implies that the user can achieve his goal accurately and completely, efficiency concerns the effort that is needed to achieve this goal. It is important to keep in mind, that effectiveness and efficiency have a large impact on the satisfaction of the user. In general satisfaction covers a rather subjective component: It is all about what the user associates with the system and whether he experiences the use of the system as positive or negative.

An important aspect of that human centred approach is the early consideration and integration of usability in the development process including the application of appropriate methods for ensuring usability. Thereby the enormous costs for staff training or for changes on the already established system can be reduced. In the worst case the consequence of the non-consideration of usability during the development of interactive systems is the development of a system, that provides several functions, but the user does not know how to use them in an effective and efficient way.

DESIGN APPROACH

Designing usable systems implies acknowledging usability as a central quality attribute from the beginning of the development and defining it as one development goal. In order to achieve this goal a systematic application of appropriate methods according to the usability engineering process as described within the DIN EN ISO 13407 (ISO 13407, 1999) is necessary. The usability engineering process is part of the development process of technical systems (Sarodnick & Brau, 2006) and complements the technological engineering process by aspects of usability.

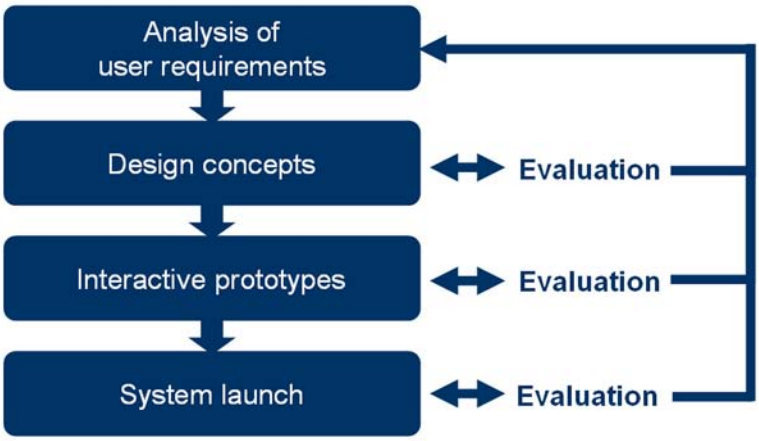
One of the typical technological engineering processes (EN 50126, 1999) defines a V-Model process which includes early verification and validation activities. The left wing of the "V" contains the specification and definition activities as well as the vertical verification. The right wing of the "V" represents the implementation and integration activities. Between both wings are the horizontal validation activities located. The V-model requires that the V & V activities start as early as possible and are continuously iterated. The amendment of the usability engineering is a consistent evolution of the idea from a methodical point of view.

The usability engineering process does not constitute a one-shot affair, where the user interface is fixed up before the release of the system (Nielsen, 1993). Instead usability engineering encompasses a set of activities throughout the development process. A huge part of these activities take place in early stages of development long before the user interface has even been designed. The application of usability engineering methods serves on the one hand the goal of achieving a good usability and on the other hand as an instrument to measure whether a good usability is achieved.

In the context of the development of interactive systems within the railway domain mainly technical aspects have been considered. The main focus usually was to provide functions within interactive systems to accomplish certain tasks. However, the provision of functions without regard to usability is only one aspect during system development. The early integration of potential system users in the development process is an essential component of the usability engineering approach (Figure 2).

Therefore the requirements out of a user perspective have to be analyzed in a systematic way at the start of the development of an interactive system. Only on the basis of such a user centred requirement analysis it is possible to design a system that meets the user's needs and considers the limitations and capabilities of the users (for example train drivers, dispatchers and signallers). Furthermore usability evaluations and corresponding adjustments of the system have to be carried out throughout the development process. This evaluation implies for example usability tests with

Figure 2. Approach to design usable systems



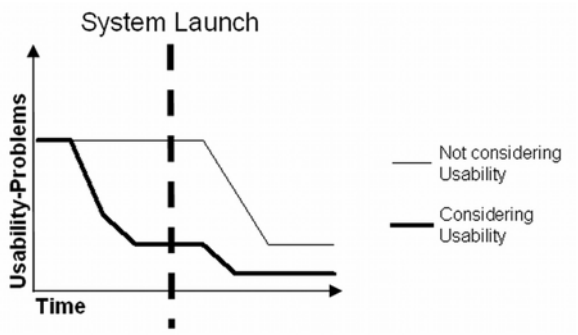
potential users or a so called expert-technique where usability experts examine the interactive system and judge its compliance with recognized usability principles. An essential aspect within this human centred development process is the iterative approach. The insights and ideas from the evaluation sessions flow directly back into the development. The evaluations can be performed already at a time when no major investment or commitment in detailed design or code has been made. One advantage of this procedure is that changes are simple and easy to make in reaction to the evaluation results. As soon as the new system is introduced to the market it will be automatically evaluated by the potential user. However usability problems discovered at this moment lead to significantly higher costs for the elimination or reduction (Heinsen & Vogt, 2003). Low usability can be handled e.g. by training the system users or by modification of the problematic aspects in the system. In general changes during early stages of development are less costly than changes made after the market introduction of a system (Rosson & Carrol, 2002). Through a systematic usability evaluation most usability problems can be detected and solved during early development phase (Figure 3).

In the following sections the steps involved in designing usable systems are described in more detail.

Analyze User Requirements

The analysis of the user requirements right at the beginning of system development is one of the main and essential tasks within the human centred development process. Through the application of usability methods the technological requirement process can be expanded by aspects of usability through a thorough elaboration of

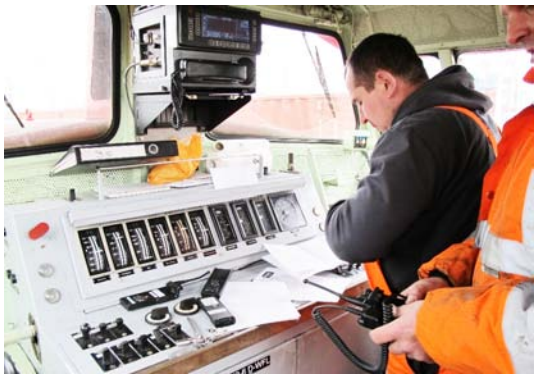
Figure 3. Not-consideration vs. consideration of usability during the development process (schematic)



the requirements out of a user perspective. One of these usability requirement methods is the so-called Contextual Inquiry. In order to develop a system which meets the user's real needs, it is essential to get a fundamental understanding of the potential users and their work practice. Since ordinary tasks become habitual and the execution is more and more unconscious over time people are not aware of all the things they do. Therefore a deep understanding of the current work practice is not accessible by simply asking users about their requirements in a regular interview. Another possibility would be to observe the users when they are conducting their tasks. But observation alone often does not give enough insight into the underlying work structure. In contrast to both briefly described methods above the Contextual Inquiry comprise semi-structured field interviews conducted in the user's workplace that focus on observation of ongoing work (Holtzblatt et al., 2005). Through this combination of observation and questioning in the real work place the actual work context, the work artifacts and the work tasks together with the reasons and implications behind the user actions can be uncovered. In the context of a development project for an interactive system for train drivers, this implies e.g. to accompany different train drivers on various runs in their cabs in order to observe and question them on their work practice in their work context (Figure 4).

Another effective method to analyze user requirements is a so-called focus group. Focus groups are moderated workshops with about 6 to 10 users in order to elicit perceptions, feelings, attitudes and ideas of the users about e.g. an existing or planned work system. The focus group has to be run by a moderator who maintains the group's focus. The application of moderation methods allows a structured collection and filtering of information. An advantage of this method lies within the group dynamic process that arises from spontaneous reactions of participants towards opinions expressed by other participants.

Figure 4. Contextual Inquiry



Thereby the discussion reveals not only new ideas on the part of users but also a deeper understanding about the underlying goals and needs of users on the part of the moderator. In order to conduct such a moderated workshop with users from the specific railway context, a basic understanding of the railway system on part of the moderators is obligatory. Otherwise the clarification of e. g. railway specific terms, abbreviations or background information would take too much time which would make an efficient and qualitative high moderation nearly impossible.

Create Design Concepts

The next step after analyzing the user requirements is to envision a first concept of the system to be developed. The aim is to produce a rough draft of the new system in which concrete screen layout or design issues are not yet taken into account. One usability method which supports the production of such a first system draft is the creation of so-called storyboards. This technique facilitates the illustration of the interaction between a system and a user in a narrative format. It includes a series of drawings, sketches or pictures and sometimes words that tell a story. A storyboard can help to illustrate and organize first design ideas. Another technique which can be used in order to create

first concept drafts is the production of so-called low-fidelity paper prototypes (Figure 5).

Both techniques are cost effective methods to make the identified requirements tangible and imaginable. They both help to clarify requirements and give a first impression of the future system. Thereby the creation of storyboards and/or paper prototypes often reveals so far undiscovered requirements. Furthermore they can be used to gather early feedback from potential users to early concept drafts through usability tests. Thereby the sketchiness of storyboards and paper prototypes is a particular benefit because it implies that the work is still in progress and indicates that there is still much space for new ideas, thus encouraging users to give feedback to really fundamental issues like structural aspects. Of course it would also be possible to create high-fidelity interactive prototypes which almost look like a real user interface (see below) already during early system development stages. A major problem with the creation of high-fidelity prototypes too early during system development is that the user's feedback on them can contain a great deal of design issues which are not of interest in that stage. Since design issues such as colors or font sizes are so obviously not the focus of such a storyboard or paper prototype these aspects are usually not commented by the users in a user feedback session. Another advantage of paper prototypes compared to high-fidelity interactive prototypes is that the feedback from the users can be directly implemented in the prototype during the evaluation session since they can be revised very quick and flexible.

Figure 5. Paper prototype



According to the iterative approach results of such user feedback sessions are directly incorporated into the revision of the prototypes, which than can be evaluated again by potential users. This loop can be repeated as often as necessary, depending on e.g. resource restrictions. Thereby the iterative procedure is more important than a large number of subjects per session.

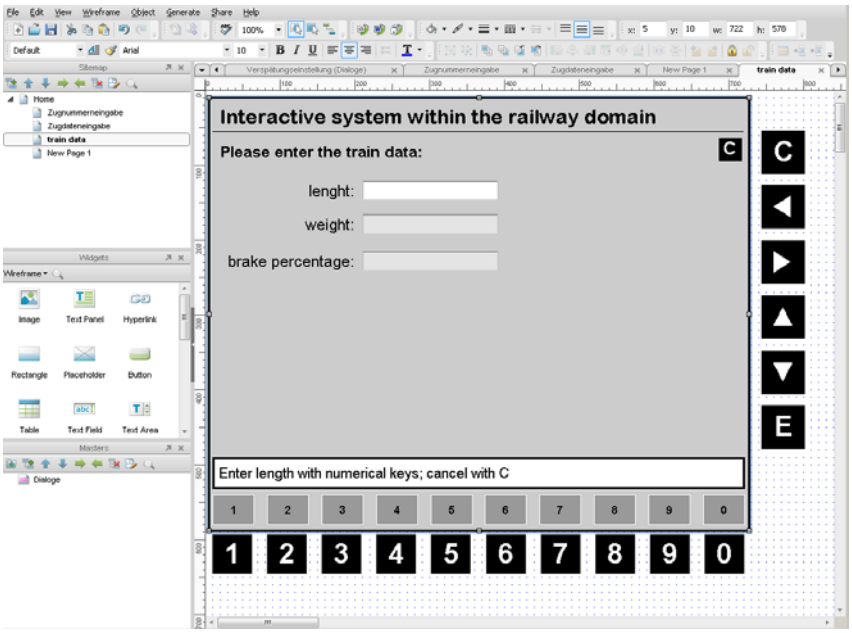
Create High-Fidelity Interactive Prototypes

After the initial design concept has been evaluated through potential users and revised in accordance with the user feedback, the next step is to create an interactive prototype of the system to be designed.

Such a high-fidelity prototype is a computer-based simulation of some but not all features of the system to be developed and it usually allows realistic user interactions within one dialogue or between different dialogues. Usually another software tool is used to mock-up the interface (Figure 6).

This software tool accepts input from the keyboard or mouse like the actual interface would, and responds to those events in the same way (displaying a particular window or message, changing state, etc.) the actual interface would respond. Therefore the concept can be evaluated through an almost realistic usability test with potential users. In contrast to low-fidelity paper prototypes high-fidelity interactive prototypes can be used to collect human performance data during usability tests (see below) like the time needed to complete a task. The results of these

Figure 6. Interactive prototype (screenshot from Axure – prototyping software)



usability tests are directly incorporated in the revision of the prototype in an iterative process until ultimately the concept for the user interface is finalized.

Evaluate Usability

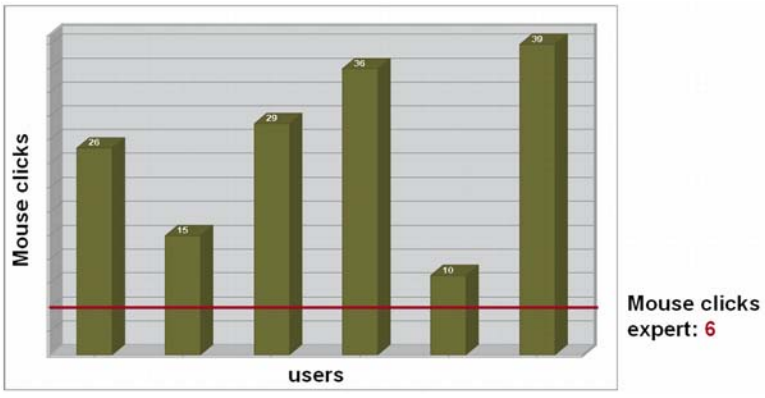
As described above the continuous iterative usability evaluation throughout the development process is a central part of the usability engineering approach. Accordingly, already the first drafts of design ideas for an interactive system to be developed have to be evaluated. But also the interactive prototypes which are created during later development stages are the subject of such a usability evaluation before the final concept can be implemented in the real system. Furthermore another important task for human factors specialists is the evaluation of already existing interactive systems in order to derive recommendations for a usable redesign. Therefore a major aspect of a usability engineer within the railway domain is the evaluation of systems to be developed or already existing interactive systems. This evaluation is done on the one hand via the application of empirical methods (e.g. user survey and/or usability tests with potential users) and on the other hand via the application of analytical methods (e.g. expert evaluation). The expert evaluation encompasses a systematic inspection of the (planned or existing) system by usability experts. The experts examine the system and judge its compliance with recognized usability principles (Nielsen, 1993). The result of such an inspection is a list of usability problems with references to the violated usability principles. On the basis of this list it is possible to derive redesign solutions for the system under inspection. Through the application of this method a large number of usability problems can be uncovered. A typical set of usability principles which can be used for such an expert evaluation are the dialogue principles within the DIN EN ISO 9241-110 (ISO 9241-110, 2006). This standard describes the following seven principles which should be taken into account when designing an interface between users and interactive systems:

- Suitability of task
- Self-descriptiveness
- Conformity with user expectations
- Suitability for learning
- Controllability
- Error tolerance
- Suitability for individualization

The dialogue principles are formulated in general terms without reference to situations of use, application, environment or technology. Therefore they can be applied for the evaluation of any interactive system. Nevertheless, an expert evaluation of interactive systems within the railway domain is of particular challenge since a basic understanding of the railway system and the respective tasks of the employees with the rail specific interactive systems is required.

In addition to an analytical evaluation through usability experts it is also possible to evaluate interactive systems or first concept drafts of planned interactive systems through empirical methods such as observation or questioning of potential users. In a so-called usability test a certain number of representative users evaluate the system in the present state of development by performing several specific task previously defined by the experimenter with the system. As stated above the 'system' is either an already existing system or a (paper respectively interactive) prototype of a future system. The quality of the results of such a test is highly dependent on the test tasks. Therefore the tasks are the beating heart of a usability test as they determine the parts of a system that the users will see and interact with – correspondingly the parts which will be evaluated. According to Lindgaard and Chattratichart (Lindgaard & Chattratichart, 2007) usability test tasks are that critical that they are even more important than the number of test users. They found out that there is no correlation between the percentage of usability problems found in such tests and the number of test users. They identified that there is a correlation between the number of identified problems and the number of user tasks tested. Hence, the development of relevant and realistic test tasks to uncover usability problems is a challenging and essential process step. During such a test the subjects are observed by usability experts who note where the users experience problems while performing the test tasks. To get insights into the mindset of the user, users are often asked to comment their thoughts as they perform the tasks (thinking aloud technique). These verbalizations provide a great opportunity to understand the expectations that users have, the mistakes that users make and to get an idea of what might be the cause for these mistakes. When conducting a usability test with

Figure 7. Number of mouse clicks as a measure for performance (screenshot from Morea – usability testing and market research software)



existing systems or interactive prototypes it is helpful to make use of specialized software tools in order to record the system interactions and a video altogether with the audio commentary of the user. These tools not only allow a detailed follow-up processing of the test, but above all, a quick analysis of performance data such as the number of required mouse clicks or time needed to perform the tasks. Figure 7 shows an example from such an automated analysis of usability test material: four out of six users needed significantly more mouse clicks to solve the task than would have been necessary.

Even though the task could be solved with six mouse clicks the last subject in the figure above needed 39 mouse clicks to solve the task. Such results are a clear indication for a usability problem. In addition to the analysis of performance data, the recorded audio commentary collected by the method of thinking aloud is also examined by the usability experts in order to understand the underlying causes of these problems.

FUTURE RESEARCH DIRECTIONS AND CONCLUSION

Interactive systems are now an indispensable part of everyday work life. Employees interact constantly with various interactive systems via a user interface without even being aware of it. Likewise the operations of interactive systems play an essential role in the ordinary work of employees within the railway domain. As mentioned above usability is a quality attribute of every interactive system - the only question is whether it is present to a greater or lesser degree. This question has also to be asked for rail specific interactive systems. The design of operating systems which are suited for use is becoming more and more important in the context of a safe and efficient rail operation, particularly in the fallback mode when automated processes need to be taken over by a human operator. In such situations it is especially important to minimize the appearance of usability problems. This chapter described a human centred approach to design usable new interactive systems but also to redesign existing interactive systems within the railway domain.

REFERENCES

Bias, R. G., & Mayhew, D. J. (Eds.). (1994). *Cost-Justifying Usability -An Update for the Internet Age*. San Francisco, CA: Morgan Kaufmann Publishers.

Dumas, J. S., & Redish, J. C. (1999). *A Practical Guide to Usability Testing*. Intelect.

EN 50126 (1999) Railway applications - The specification and demonstration of reliability, availability, maintainability and safety (RAMS)

Heinsen, S., & Vogt, P. (Eds.). (2003). *Usability praktisch umsetzen – Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte* [Usability into practice - Manual for software, web, mobile devices and other interactive products]. München: Karl Hanser Verlag. In German

Holtzblatt, K., Burns Wendell, J., & Wood, S. (2005). *Rapid Contextual Design – A How-To Guide to Key Techniques for User-Centered Design*. San Francisco, California: Morgan Kaufmann Publishers.

ISO 13407 (1999)*Human-centered design processes for interactive systems* .

ISO 9241-110 (2006)*Ergonomics of human-system interaction - Part 110: Dialogue principles*.

Jordan, P. W. (2001). *An Introduction to Usability*. Philadelphia, PA: Taylor & Francis.

Kuniavsky, M. (2003). *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco, CA: Morgan Kaufmann Publishers.

Lindgaard, G., & Chattratichart, J. (2007). Usability testing: what have we overlooked? In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. (pp. 1415-1424). New York, NY: ACM.

Nielsen, J. (1993). *Usability Engineering*. San Diego, California: Academic Press.

Norman, D. A. (2002). *The design of everyday things*. New York: Basic Book.

Raskin, J. (2000). *The humane interface – New directions for designing interactive systems*. Crawfordsville, IN: ACM.

Rosson, M. B., & Carrol, J. M. (2002). *Usability Engineering – Scenario-Based Development of Human-Computer Interaction.* San Francisco, California: Morgan Kaufmann Publishers.

Sarodnick, F., & Brau, H. (2006). *Methoden der Usability-Evaluation - Wissenschaftliche Grundlagen und praktische Anwendungen* [Methods of usability evaluation - scientific fundamentals and practical applications]. Bern: Hans Huber. In German

ADDITIONAL READING

Snyder, C. (2003). *Paper prototyping – the fast and easy way to design and refine user interfaces*. San Francisco, CA: Morgan Kaufmann Publishers.

KEY TERMS AND DEFINITIONS

Contextual Inquiry: An empirical method for the analysis of requirements out of a user's perspective.

Expert Evaluation: An analytical usability evaluation method in which usability experts examine an interactive system and judge its compliance with recognized usability principles.

Paper Prototype: Fast and rough creation of the layout of a low fidelity interactive system by using paper and cutting and pasting the single elements.

Usability: Usability is a quality attribute of every interactive system.

Usability Test: An empirical usability evaluation method in which potential users evaluate an interactive system through the performance of specified test tasks.

User Interface: The language through which the system and the user interact with each other.

Chapter 15
Integration of Human Factors to Safety Assessments by Human Barrier Interaction

Markus Talg
German Aerospace Center, Institute of Transportation Systems, Germany

Malte Hammerl
German Aerospace Center, Institute of Transportation Systems, Germany

Michael Meyer zu Hörste
German Aerospace Center, Institute of Transportation Systems, Germany

ABSTRACT

Human factors have a strong impact on railways safety. However, the assessments of these factors still follow traditional and inadequate approaches. While failure probabilities of technical systems can be measured in sufficient precision, human error probabilities are still estimated in a very rough and vague way. Upon this motivation, the contribution presents a method analyzing human influence in railway applications. The approach of human-barrier-interaction relies on a new model of human behavior, a classic model of human-machine-interaction and a model of safety measures by barriers. Applying the method, human reliability can be assessed in comparative way. An advantage over existing approaches is the substantial combination of cognitive psychology and engineering expertise without unpractical complexity.

INTRODUCTION AND BACKGROUND

The European standards on railway safety request a risk oriented approach assessing hazards within the railway system. These analyses need data not only on the reliability of technical systems but also of human factors. The rail specific standard EN 50126-1 (CENELEC, 1999) as well as the European regulation on "Common Safety Methods" (EC 352/2009) prescribe a detailed human factors analysis in risk assessment. The methods used up to now rely on old-fashioned or even outdated approaches and are therefore inadequate in coping with the requested integration of human factors.

DOI: 10.4018/978-1-4666-1643-1.ch015

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

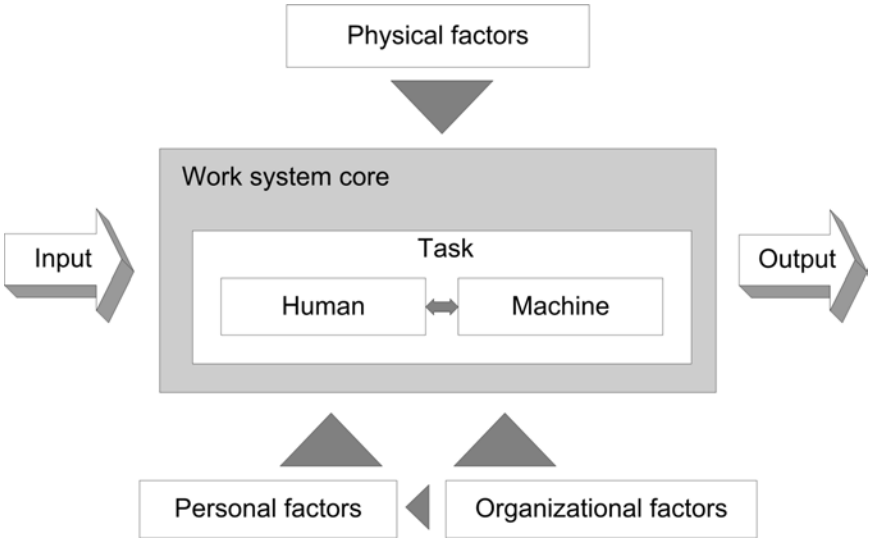
Often, engineers still follow the classic maxims of continuous automation or protection of human actions by technical systems. In spite of a high level of automation and protection, humans still bear responsibility for railway safety. This particularly goes for disrupted modes of operation when technical protections are deactivated. Continuous automation involves problems as not all human operations can be replaced by technical systems. Due to the automation the operator becomes unfamiliar with a lot of formerly well-known tasks. In case of technical disturbances, suddenly the operator has to fulfill these rarely practiced tasks. This can directly lead to work and stress overload and finally to an augmented error proneness. Another possibility increasing the safety level is the use of technical systems as safeguards: the system reacts in times operators do not fulfill their tasks correctly. The engineering can be very costly in this case. A third traditional approach is the gradual extension of instructions, which tends to result in complex and incomprehensible rules and standards. This fact rather provides new starting points for different human errors. In any way, the human remains in the system with a non-negligible responsibility and the need for reliable statements about the influence of human factors in risk analyses persists.

In European railway engineering practice, often the fixed error probability 10^{-3} is chosen. But due to the variability of human behavior, fixed human error probabilities do not model the human impact in an adequate way. For many human actions the fixed value 10^{-3} seems to be very conservative witch lead to oversized system designs. More sophisticated analyses use a set of 18 fixed error probabilities published by Hinzen in German literature (Hinzen, 1993). These 18 fixed values vary in dependence of stress level, surrounding conditions and the human information process. Hinzen's model is predicated to failure rates from other industries and obsolete human-machine-interfaces. Furthermore performance shaping factors can't be integrated. In the latter study, human behavior is classified along of stress and difficulty levels. Studies on working places at nuclear plants in the 1980s served as background of this principle. Over the years, the working profiles of operating staff in particular in the railway domain has changed intensively so that the comparability to 1980s working places is not given. So, neither the fixed value of 10^{-3} nor the set of values by Hinzen are suitable approaches.

In other industry domains, a lot of energy has been spent in research on human reliability assessment (HRA). Particularly to the so-called first generation of HRA methods, a certain criticism has been raised in literature. Methods are said not to include all error types and not to integrate the operator's goals when performing a task (Sträter, 2005). The only prospective railway specific HRA-method is called rail-HEI/rail-HEQ (Human Error Identification/ Quantification) and was published by the British Rail Safety & Standard Board (RSSB, 2004). Rail-HEQ is based on the "Human Error Assessment and Reduction Technique – HEART" (Williams, 1986). Like HRA-methods designed in other industries rail-HEI/rail-HEQ is very complex and presupposes expertise in cognitive psychology. Unfortunately, the method has consistency issues between the two parts error identification and error quantification (Hickling, 2007). The second generation of HRA, generally, integrates the phenomenon of *errors of commission*: human actions that are not required from a system's point of view and aggravate the scenario's evolution. But, the high complexity of HRA methods remains a problem. Due to difficulties in obtaining reliable data, the validity stays questionable, particularly for human error quantification in railways.

One reason that human error assessment remains not perfectly solved so far is the lack of suitable data to validate theory. Incident databases are not always accessible and subject studies in simulation environments are time-consuming.

Figure 1. Work system of human-machine-interaction



Finally, expert judgments are subject to criticism being too subjective. Additionally, the first and classic approaches to model human error involved certain simplifications that have not proven suitable for engineering practice.

Unfortunately, the use of error taxonomies has not changed since the phenomenon of errors of commission was analyzed more deeply. For example, the classic scheme by James Reason (Reason, 1990) (slips, lapses, errors etc.) persists although it neither integrates errors of commission nor distinguishes between phenotypes (observable error types) and genotypes (error evolution) of human error (Hollnagel, 1993).

As a conclusion, a manageable tool to integrate human factors into railway safety assessments was missing so far. Particularly, it lacks of a reliable description of human behavior. The remainder of the chapter presents the human-barrier-interaction, a straight-forward approach to include the assessment of human factors in railway applications. So far the method allows a comparative assessment of different barriers with the same function. In further research the method has to be extended to serve semi quantitative statements.

ASSESSMENT OF HUMAN FACTORS WITH HUMAN-BARRIER-INTERACTIONS

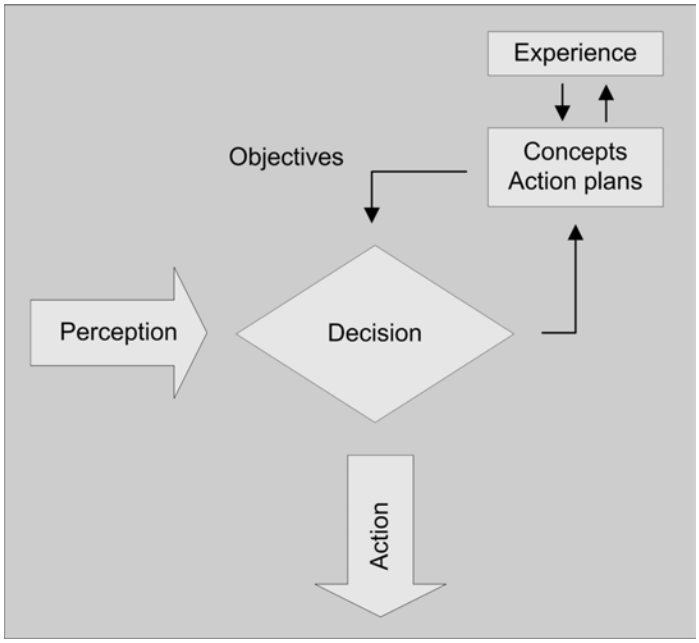
Basic Settings for the Human-Barrier-Interaction

This section presents the scientific models on which the analysis is based. The approach "human-barrier-interaction" relies on a classic model of human-machine-interaction, the modeling of safety measures by barriers and a tabular model of human behavior.

First foundation for the approach is a bilateral model of human-machine-interaction. The interaction in the "work system core" takes place between the operator and the technical system (Figure 1). On the horizontal axis, the work system in its dynamic operation receives inputs and generates outputs. The vertical influences are less dynamic set variables for the work system. It was proposed to structure performance shaping factors into physical, personal and organizational factors.

Figure 2 shows an integrated, mental model of human decision processes and possible errors

Figure 2. The cognitive processing loop © 2005 Oliver Sträter. Used with permission



(Sträter, 2005). In contrast to very classic (and mostly sequential) approaches, the decision process between perception and action is understood as a dynamic circle with the contribution of concepts, experience and objectives. This model overcomes (but still integrates) the classic triple perception, cognition, action.

For risk assessments, the idea of barriers has been studied frequently. Barriers are defined as safety mechanisms that are installed to prevent undesired events from taking place or to protect against its consequences (Sklet, 2006). The most common taxonomy of barriers distinguishes between physical, functional, symbolic and immaterial barriers (Hollnagel, 2008). Physical barriers physically exist in the system while functional barriers create physical or logic dependencies. Examples for symbolic and immaterial barriers are signs and rules respectively. Additionally, a three-part categorization has been proposed: barriers of prevention that prevent an undesired initial event from taking place, barriers of correction that recover the situation and barriers of containment that lessen the severity of the consequences (also see Sklet, 2006). Furthermore, a relationship between the barrier system and the barrier function can be found in literature (Hollnagel, 2004). Barrier systems describe the means by which the barrier functions are carried out. While physical and functional barrier systems perform their barrier functions themselves, symbolic and immaterial barriers request an action and its execution finally represents the barrier function.

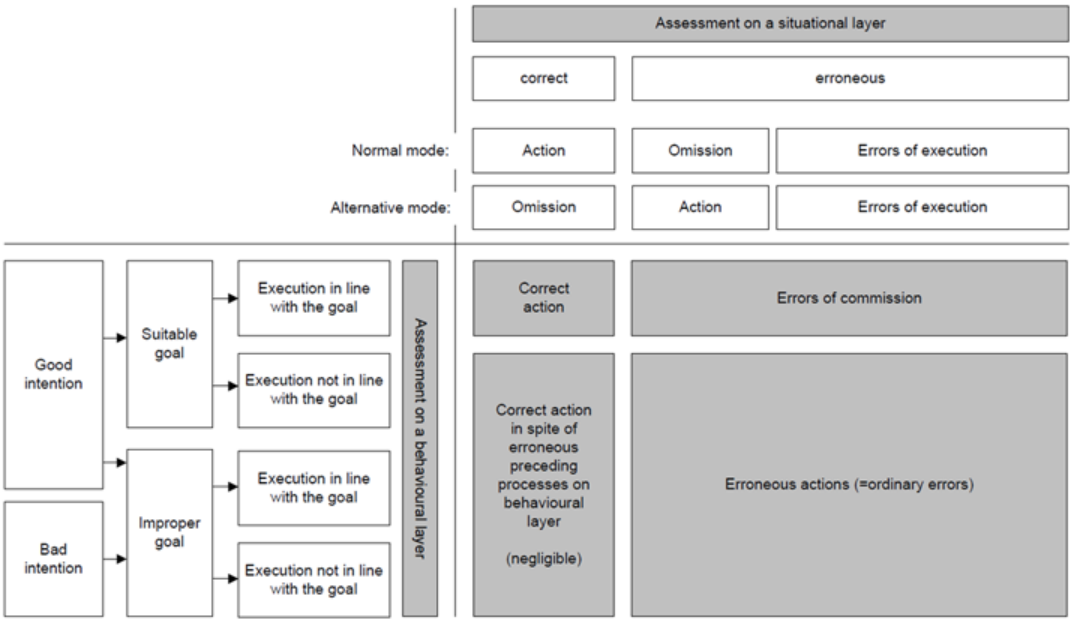
For the assessment of the efficiency or quality of barrier, different sets of criteria have been proposed. But, human interaction is not always listed or is only represented in a one-dimensional way "dependency on human actions". In this context, it has been stated that – in contrast to physical and functional barriers – symbolic and immaterial barriers can generally be estimated as being medium or less efficient (Hollnagel, 2008). Concerning human influence on the barrier function, a lot of studies have been presented with the focus set to barrier removal or barrier crossing (e.g.

Polet, 2002). Mostly, these works are limited to intentional deviations. A holistic perspective on the interaction of humans with barriers is missing. Given the popularity of barriers and the unsolved precise assessment of human reliability, an analysis of the human impact on barrier functioning seems promising.

One problem of classic taxonomies of human reliability is the missing integration of *errors of commission* and the missing distinction between phenotypes and genotypes. These issues can be handled with a tabular model of human behavior (Hammerl, 2011). An error of commission occurs when the operator might assume a certain action to be right although the action is not correct in the certain situation. This can be the case if the machine is in another state than the user suspected. So, the distinction between phenotype and genotype is similar to a judgment on the operator's point of view (behavioral layer) and a system's point of view (situational layer). Putting the two layers in one chart, a tabular model of human behavior results (see Figure 3).

Most normally, actions judged correct by the operator are correct on the situational layer. If they are not, an error of commission is recorded. In this case, the action is not required from a system's point of view. When an operator fails in doing the right thing, the action is normally also incorrect on the situational layer. However improbable but possible, the action can also be correct when the situation is different to what is expected. Errors of execution are actions that are incorrect in at least one dimension. Examples are "too early/late", "too much/less" etc. The distinction between correct and faulty actions on the situational layer is made according to a certain threshold. Note that omission can be the correct action in some cases, called alternative modes, here. This tabular model distinguishes between two dimensions of judging human behavior. It depicts errors of commission. The model shows that errors on one layer cannot be equated with errors on the other layer. Together with human-machine-interaction and the theory of barriers, the model serves as a basis for the approach of "human-barrier-interaction".

Figure 3. Tabular model of human behavior



Assessment of Human-Barrier-Interaction

The interaction between the operator and the technical system is modeled as a bilateral human-machine-interaction. In this interaction, safety critical tasks are performed. The central questions of "human-barrier-interaction" are:

- When the operator has an impact on the safety function of a barrier, by which human-machine-interaction is this safety function fulfilled?
- How reliable is a barrier where the human has an impact on the safety function?

The main thesis of the human-barrier-interaction is, *the higher the complexity of the human-machine-interaction, the higher the human error-proneness*. The steps that are necessary to perform the safety function have to be analyzed in detail in order to obtain information about the level of complexity of an interaction.

Before the assessment of human-barrier-interaction can start, naturally, the barriers have to be identified. Scanning fault trees and event trees provides the practitioner with a set of technical barriers. This procedure takes a very functional perspective on the human contribution. The barrier identification should be completed by the barrier search in so called Hierarchical Task Analyses (Annett, 2003). A bottom task in the analysis tree either fulfils a safety function or a performance function. A safety function is a function that is needed to fulfill a safe operation. If the function is not executed a hazardous situation arises. A performance function is needed to fulfill system's operations. A non-execution of this function downsizes the performance but does not lead to a hazardous situation. On the contrary, a mistimed performance function (error of commission) often has an impact on the system safety.

Example 1. "Stop the train in front of a stop signal" is a safety function. "Open the doors for entry and exit of passengers at a station" is a performance function.

Safety functions often lead to symbolic or immaterial barriers while performance functions usually lead to functional barriers. Physical barriers can be neglected because their safety function works without human influence during runtime. The barrier identification by the help of fault and event trees *and* task analyses avoids a tunnel-perspective on human errors in a single scenario, but ensures a holistic perspective on a certain context.

A distinction between barrier systems and barrier function is introduced as barrier decomposition. Thereby barrier systems describe the means by which the barrier functions are carried out.

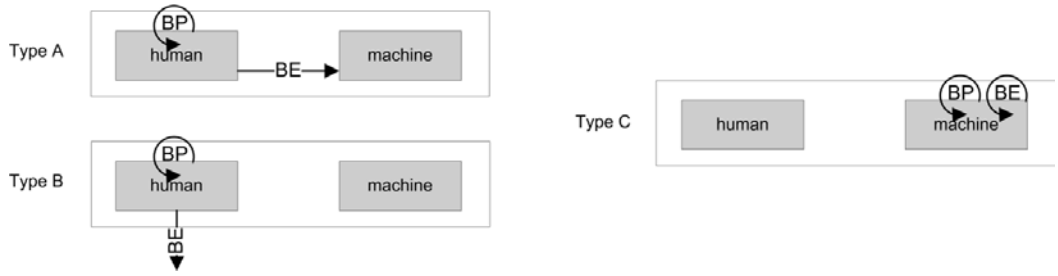
Example 2. A restrictive speed signal represents the barrier system while the triggered braking of the train driver is the barrier function. Additionally, it is to say that barriers can involve more than one barrier system.

In order to analyze the barrier function in a detailed manner, a further breakdown into three steps is proposed:

- Barrier initiation (BI)
- Barrier processing (BP)
- Barrier execution (BE)

Barrier initiation represents the introductory function of a situation or a barrier system that results in the impulse that triggers the barrier processing. In most cases the initiation covers an information retrieval and processing up to problem awareness. The decision how to act or react can be described as barrier processing. Finally, the barrier execution is the action that implements

Figure 4. Basic types of human-barrier-interactions



the barrier function. The barrier processing and the barrier execution are performed either by an operator or by a technical system. In case the operator performs the processing and the execution the barrier is symbolic or immaterial in terms of the classic taxonomy. The difference between symbolic and immaterial only depends on the way of initiation. Are barrier processing and execution performed by the technical system, the barrier is classified as functional. In this case the operator has a supervising task in monitoring the technical system and the only way of intervention is the deactivation of the barrier function. Processing and execution function are performed by the technical system, but the operator can deactivate the safety function by pushing a button to release the train out of the braking curve restriction

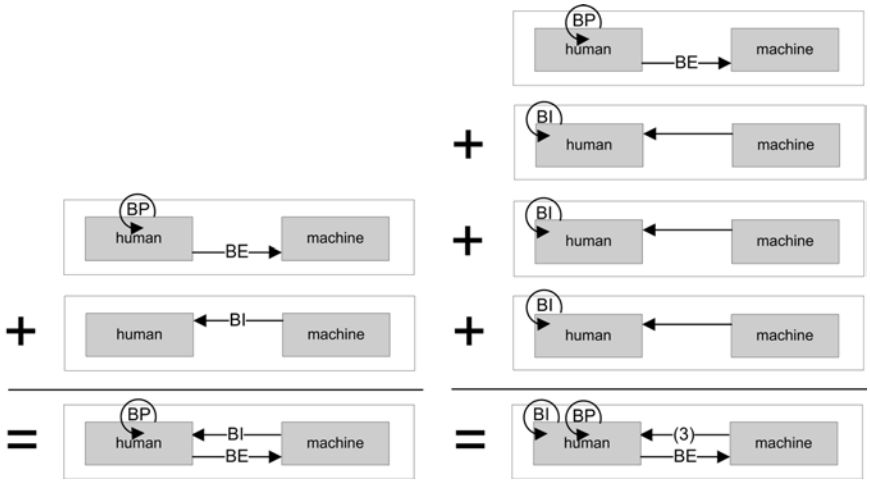
Example 3. A restrictive speed signal is a symbolic barrier. The perception of the signal presents the barrier initiation, the in mind process of the train driver that he has to reduce the speed by braking presents the barrier process and the executed breaking presents the barrier execution.

An example for an immaterial barrier is a stop order by the train dispatcher and a functional barrier is the braking curve of an intermittent train control that prevents passing a stop signal.

A combination of barrier processing and barrier execution with the classic human-machine-interaction results in three basic types of human-barrier-interactions (Figure 4). The types A and B where processing and the execution are performed by the operator represent symbolic and immaterial barriers. The processing is labeled by a circle and presents a pass trough the human information process (see Figure 2). The difference between type A and B lies in the way of execution. In type A, the human performs the execution on the technical system while in type B he performs the execution to an objective outside the human-machine-system, e.g. he gives a message to another person. Type C where barrier processing and execution are performed by the technical system represents functional barriers.

The basic types of human-barrier-interaction comprising barrier processing and execution have to be complemented by the barrier initiation. In type A and B, the barrier initiation is directed to the operator and emanates either from the machine, from outside the system or from the operator himself. Furthermore, the initiation can be active or passive. An active initiation directly requests the operator to perform an action while, in a passive initiation, the operator has to derive the problem by passing the cognitive information process.

Figure 5. Examples for human-barrier-interactions of type A, active and passive initiation



Example 4. A stop signal actively initiates the train driver to perform an action (Figure 5, left-hand side). The initiation of the task "observation of the speed limit" is passive because the driver's intention builds the initial point. In addition the train driver has to combine the information of the speed indicator, the electronic timetable display and the track mileage to realize the problem (Figure 5, right-hand side).

In human-barrier-interactions of type C, the safety function is performed by the technical system. The operator either initiates the technical system to perform the safety function (C-I) or he has the possibility to deactivate the safety function (C-II). In type C-I, a human error in a performance function initiates the safety function. Those errors in performance functions can be assessed similar to errors in safety functions like in types A and B. Please note that an error of a performance function does not immediately lead to a hazardous situation, here. The hazard only occurs when both human and technical system fail. A deactivation (C-II) only leads to a hazardous situation, when the operator deactivates the safety function in a situation the function is in fact needed.

Example 5. The omission of pushing the button or the pedal of the dead man's device is an initiation of a safety function (Figure 6, left). The release out of a braking curve of an intermittent train control system in front of a stop showing signal is a deactivation of a safety function.

A combination of the three barrier types A/B, C-I and C-II and the tabular model of human behavior allows a differentiation between classic human error and errors of commission. A classic human error in type A/B leads to a failure in the safety function while an error of commission leads to an execution of a safety function when it is not necessary. So, in type A/B a classic human error impacts a safety risk while an error of commission does not impact the safety but possibly the performance. The impact of classic human errors and errors of commission are presented in Table 1, for all barrier types. The most important barriers in terms of human influence remain symbolic and immaterial barriers (types A and B). In other cases, there is still a certain level of technical protection. The error-proneness of symbolic and immaterial barriers can be assessed by the complexity of the human-machine-interaction.

Table 1. Comparison of barrier types related to safety their function and possible error types

Barrier type	A/B	C-I	C-II
Classification	symbolic or immaterial	functional	functional
Execution of safety function	The operator executes safety function	Technical system executes safety function when the operator fails	Technical system executes safety function. The operator can deactivate safety function
Occurrence of hazard	Human error leads to a hazard	Human error and failure in technical system lead to a hazard	Error of commission or failure in technical system leads to a hazard
Classic human error (failure in behavior and situation layer)	No execution of safety function (safety risk)	Technical system reacts of human error (no safety risk, possible loss of performance)	Mistimed execution of safety function (no safety risk, possible loss of performance)
Error of commission (failure in situation layer)	Mistimed execution of safety function (no safety risk, possible loss of performance)	No effect (no safety risk)	Mistimed deactivation of safety function (safety risk)

The assessment of the complexity for a human-barrier-interaction will be done on the basis of a set of criteria which have its seeds in engineering psychological principles: the higher the difficulty to understand and to process information, the higher the error-proneness. The criteria for a higher complexity are:

- Passive initiation versus active initiation
- Number of barrier systems the operator has to interact with
- Number of passes through the cognitive human information process
- Spatial distance between barrier system and barrier function
- Temporal distance between barrier system and execution of the barrier function

The presented method assessing human errors will be illustrated with the help of an example: two barriers fulfilling the same safety function will be compared. On German railway tracks with intermittent train control system the train driver has to observe the track speed in monitoring actual speed on the speed indicator the actual position on the track mileage on track side and the permitted maximum speed on the electronic timetable (see Figure 5). The train driver has to combine these information sources to adjust an adequate speed. In control level 2 of the new European train control system ETCS the train driver gets all information on one display. It shows the actual speed and the actual permitted maximum speed on the speed indicator. A speeding can be easily detected because in this case the pointer colors orange. The color change initiates the train driver actively. The human-barrier-interaction for the speed observation in intermittent train control equals the right side and the observation in ETCS Level 2 equals the left side of Figure 5. A comparison of the barriers on basis of the above mentioned criteria reveals that the complexity observing the track speed on tracks with intermittent train control system is much higher than on tracks with ETCS Level 2 (see Table 2).

Consider the thesis *the higher the complexity of the human-machine-interaction, the higher the human error-proneness:* observing the track speed on ETCS Level 2 tracks is more reliable than observing the speed on tracks with intermitted train control system.

With the set of criteria, a solid and handy solution for the comparative assessment of different barriers is available. For the moment, the focus

Figure 6. Examples for human-barrier-interactions of type C-I (operator initiates and C-II (operator deactivates)

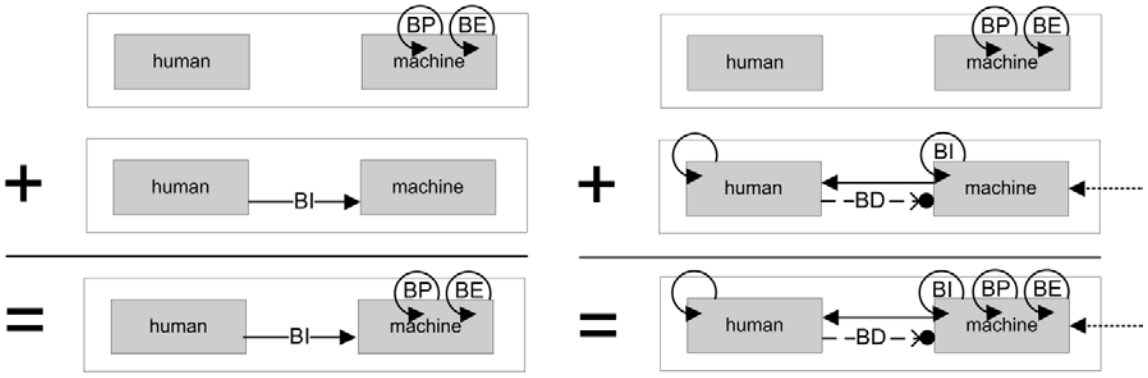


Table 2. Example observing track speed comparing different barriers with the same intention

Barrier: Speed observation in …	Intermittent train control system	ETCS Level 2
Barrier type	A	A
Initiation	passive	active
Number of barrier systems	3	1
Number of passes through the cognitive information process	4	2
Spatial distance	yes	no
Temporal distance	yes	no

is set to barriers where the operator performs the safety function. The tabular model of human performance and the table of risk allocation (Table 1) enable an overview on the influence of human error on system safety.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In the contribution, the integration of human factors into safety assessments was discussed. A method evaluating the impact of human factors on railway safety was presented. The human-barrier-interaction allows the user to compare the reliability of different human tasks or safety measures fulfilling the same purpose. The comprehensible visualization make the method manageable for safety engineers while the psychological expertise is practically integrated by criteria in human-machine-interaction. The distinction between phenotypes and genotypes of human error considering the operator's point of view on the one hand and the system's point of view on the other hand enables an assessment of errors of commission.

All in all, the human-barrier-interaction provides a basic procedure for integration of human factors in safety and reliability considerations as requested in the European standard EN 50126-1. A consideration of human and technical performance with a holistic perspective on the performance of safety functions was proposed. In further research the qualitative method for the comparative assessment of human reliability has to be expanded to a semi-quantitative method by defining safety

levels that define numerical intervals for the human error proneness. This classification has to be validated with psychological studies, interviews with safety experts and of course with a database of real incidents. Furthermore performance shaping factors for the operator have to be integrated to assess the reliability in specific situations.

REFERENCES

Annett, J. (2003): Hierarchical Task Analysis. In: LeBlanc Dobson, D. (Author); Hollnagel, E. (ed.): *Handbook of Cognitive Task Design*, (pp.17-35) Lawrence Erlbaum Associates

CENELEC (1999). EN 50126-1 Railway applications - The specification and demonstration of Reliability, *Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process. Corrigendum 2006*.

*EC 352/2009 (2009).*COMMISSION REGULATION (EC) No 352/2009 of 24 April 2009 on the adoption of a common safety method on risk evaluation and assessment as referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council

Hammerl, M. (2011) *Analyse der menschlichen Einflussfaktoren und Zuverlässigkeit im Eisenbahnverkehr. (Analysis of human factors and reliability in railways)*. PhD thesis. TU Braunschweig, Germany.

Hinzen, A. (1993). *Der Einfluss des menschlichen Fehlers auf die Sicherheit der Eisenbahn (The influence of human error on safety in railway applications)* (*Vol. 48*). Aachen, Germany: Veröffentlichung des Verkehrswissenschaftlichen Instituts der RWTH Aachen.

Hollnagel, E. (1993). *Human reliability analysis: Context and control*. London: Academic Press.

Hollnagel, E. (2004). *Barriers and Accident Prevention*. Hamshire, England: Ashgate Publishing Limited.

Hollnagel, E. (2008). Risk + Barriers = Safety? *Safety Science*, *46*, 221–229. doi:10.1016/j.ssci.2007.06.028

Polet, P., Vanderhaegen, F., & Wieringa, P. A. (2002). Theory of safety-related violations of system barriers. *Cognition Technology and Work*, *4*, 171–179. doi:10.1007/s101110200016

Reason, J. (1990). *Human Error*. New York, USA: Cambridge University Press.

RSSB. (2004). *Rail-Specific Human Reliability Assessment Technique for Driving Tasks, T270 Final Report*, Rail Safety and Standard Board,Retrieved from: www.rssb.co.uk.

Sklet, S. (2006). Safety barriers: Definition, classification, and performance. *Journal of Loss Prevention in the Process Industries*, *19*, 494–506. doi:10.1016/j.jlp.2005.12.004

Sträter, O. (2005). *Cognition and Safety – An Integrated Approach to System Design and Assessment*. Hampshire, England: Ashgate Publishing Limited.

Vanderhaegen, F., Polet, P., Zhang, Z., & Wieringa, P. A. (2002). Barrier removal study in a railway simulation, *PSAM 6,* Puerto Rico, USA.

VDI. (2010). *4006 Part 3: Human reliability - Methods to analyse events regarding human behaviour, draft*. Düsseldorf, Germany: VDI-Gesellschaft Produkt- und Prozessgestaltung.

Williams, J. C. (1986). A proposed Method for Assessing and Reducing Human error. *Proceedings of the 9th Advance in Reliability Technology Symposium. pp. B3/R/1-B3/R/13*, University of Bradford

ADDITIONAL READING

Baysari, M., Caponecchia, C., McIntosh, A., & Wilson, J. R. (2009). Classification of errors contributing to rail incidents and accidents: A comparison of two human error identification techniques. *Safety Science*, *47*(7), 948–957. doi:10.1016/j.ssci.2008.09.012

Belmonte, F., Boulanger, J.-L., & Schön, W. (2008): Human reliability analysis for automatic train supervision. *10th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Human-Machine Systems*, Korea.

Feldmann, F., Hammerl, M., & Schwartz, S. (2008): Questioning human error probabilities in railways. *3rd IET International Conference on System Safety*, Birmingham.

Hale, A., & Heijer, T. (2006): Is Resilience Really Necessary? The Case of Railways. In: Hollnagel, E.; Woods, D.; Leveson, N.: *Resilience engineering: concepts and precepts*. (pp. 125-147) Ashgate publishing.

Hammerl, M., & Vanderhaegen, F. (2009): Human factors in the railway system safety analysis process. *3rd International Conference on Rail Human Factors*, Lille, France.

Hickling, N. (2007): An Independent Review of a Rail-specific Human Reliability Assessment Technique for Driving Tasks. Report T270 of the Rail Safety and Standards Board, Retrieved from: http://www.rssb.co.uk/pdf/reports/research/T270_review_final.pdf, Revision 2007-08-14 (File 2007-11-29).

Kim, D.S., & Baek, D:H.; Yoon, W.C. (2010). Development and evaluation of a Computer-Aided System for Analyzing Human Error in Railway Operations. *Reliability Engineering & System Safety*, *95*(2), 87–98. doi:10.1016/j.ress.2009.08.005

Rail Safety and Standards Board (Ed.). (2011): *Human Factors Research Library*, Retrieved from: http://www.rssbhumanfactorslibrary.co.uk.

Reason, J. (1997). *Managing the Risks of Organizational Accidents*. Ashgate publishing.

Reason, J.; Hollnagel, E.; Paries, J. (2006): *Revisiting the "Swiss Cheese" Model of Accidents*. Eurocontrol Report.

Sheridan, T. (2002). *Humans and Automation – System Design and Research Issues*. John Wiley & Sons.

Stanton, N. A., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2006). *Human Factors Methods: A Practical Guide for Engineering and Design*. Ashgate.

Sträter, O., Dang, V., Kaufer, B., & Daniels, A. (2004). On the way to assess errors of commission. *Reliability Engineering & System Safety*, *83*(2), 129–138. doi:10.1016/j.ress.2003.09.004

Traub, P. (2004): Human Error Analysis in Railway Safety Cases; Panacea or Poison? *Proceedingsof the Human Factors and Ergonomics Society Annual Meeting, 2004, Perception and Performance,* S. 2065-2069.

Vanderhaegen, F. (2001). A non-probabilistic prospective and retrospective human reliability analysismethod – application to railway system. *Reliability Engineering & System Safety*, *71*(1), 1–13. doi:10.1016/S0951-8320(00)00060-0

Wilson, J. R., Norris, B., Clarke, T., & Mills, A. (Eds.). (2005). *Rail Human Factors: Supporting the Integrated Railway*. Ashgate Publishing.

Wilson, J. R., Norris, B., Clarke, T., & Mills, A. (Eds.). (2007). *People and Rail Systems: Human Factors at theHeart of the Railway*. Ashgate publishing.

Young, M. S., Stanton, N. A., & Walker, G. H. (2006). In loco intellegentia: human factors for the future European train driver. *International Journal of Industrial and Systems Engineering*, *1*(4), 485–501. doi:10.1504/IJISE.2006.010388

KEY TERMS AND DEFINITIONS

Barrier: Safety mechanism that is installed to prevent undesired events from taking place or to protect against its consequences.

Error of Commission: Human action that is not required from a system's point of view and often leads to a hazardous situation.

Human-Barrier Interaction: Similar to a human-machine-interaction but the interaction serves for system safety.

Human-Machine Interaction: Bidirectional interaction between human and machine. Classical interactions are state information from the machine and as reaction an execution on the machine by the operator.

Performance Shaping Factors: Physical, personal or organizational factors that decrement or improve human performance.

Section 7

Security, Monitoring and Surveillance

Chapter 16
Advanced Techniques for Monitoring the Condition of Mission-Critical Railway Equipment

Clive Roberts
University of Birmingham, UK

Joe Silmon
University of Birmingham, UK

ABSTRACT

This chapter provides an overview of advanced techniques for monitoring the condition of mission-critical railway assets. The safe operation of railways depends on a large number of geographically distributed components, each of which has a low cost when compared to the highly complex arrangements of assets found in other industries, such as rolling mills and chemical plants. Failure of any one of these components usually results in a degradation of service in order to maintain safety, and is thus very costly to modern railway operators, who are required to compensate their customers when delays occur. In this chapter, techniques for industrial condition monitoring are reviewed, highlighting the main approaches and their applicability, advantages, and disadvantages. The chapter first makes some basic definitions of faults, failures, and machine conditions. The analysis of faults through methods such as Fault Tree Analysis and Failure Modes Effects Analysis are examined. The field of fault diagnosis is then reviewed, partitioning into the three main areas: numeric/analytical models, qualitative models, and data/history-based methods. Some of the key approaches within each of these areas will be explained at a high level, compared, and contrasted.

DOI: 10.4018/978-1-4666-1643-1.ch016

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

1. INTRODUCTION

Previous work on railway asset condition monitoring is examined. A railway based case study is presented, based on the results of a previous industry/academia collaborative project. The challenges for future research and development are discussed, addressing such topics as information integration, limitations of automatic monitoring and human-machine interaction.

2. A REVIEW OF INDUSTRIAL CONDITION MONITORING TECHNIQUES

2.1 Condition Monitoring Definitions and Capabilities

For the purposes of asset condition monitoring, a failure is defined as any unplanned event which results in an asset being unavailable for its designed purpose.

A fault is a condition of an asset which causes it to deviate from its normal behaviour. This may or may not cause a failure, depending on both the severity of the fault and the functional area of the asset affected by the fault. For example, a loose screw on a luggage shelf may cause rattling in the saloon of a passenger vehicle; this is a fault, because the components are behaving abnormally, but it will not necessarily cause a failure, because it does not stop the train from functioning properly or safely.

An incipient fault is a fault which develops gradually over a period of time. An example might be the loosening of non-threadlocked bolts or screws when subjected to prolonged vibration. Incipient faults are difficult to detect early using automatic methods, and therefore have a tendency to cause failures or only be detected at a late stage of development.

In condition monitoring, the terms fault detection, fault diagnosis and fault identification have distinct meanings. Detection means determining purely whether or not a fault is present in an asset. Diagnosis means to decide what fault may be present. Identification means to determine the severity of the fault.

These capabilities can be added in layers to provide progressively higher levels of condition monitoring. As further layers are added, automated systems can take on more responsibility and gradually improve asset availability by reducing time spent in reactive maintenance, diagnosis and by optimally scheduling maintenance tasks.

The flow diagrams in Figure 1 show how tasks can be split between human maintainers and automated condition monitoring systems for five levels of capability which have been defined as part of the INNOTRACK project (Silmon 2009). Table 1 categorises the capabilities of the condition monitoring system shown in Figure 1.

Currently the most advanced condition monitoring systems available for railway applications have capability level 2. The capability depends partly on the availability of a suitable algorithm or model for the asset being monitored, and partly on the opportunity for analysis or training of the models, which must be done in co-operation with the asset manager.

2.2 Fault Diagnosis Methods

Automated fault diagnosis has been an area of high research activity for several decades. Most of the effort in this field has been focussed on industrial processes, such as may be found in the chemical and manufacturing industries. Industrial processes usually have dozens or hundreds of variables and a large, expensive plant which is built as a one-off exercise. This means that investment in fault diagnosis has been higher, because it is a proportionally smaller cost to the operators and yet yields a high return in increased reliability.

Figure 1. Capability levels and the interaction between maintainers and automated systems

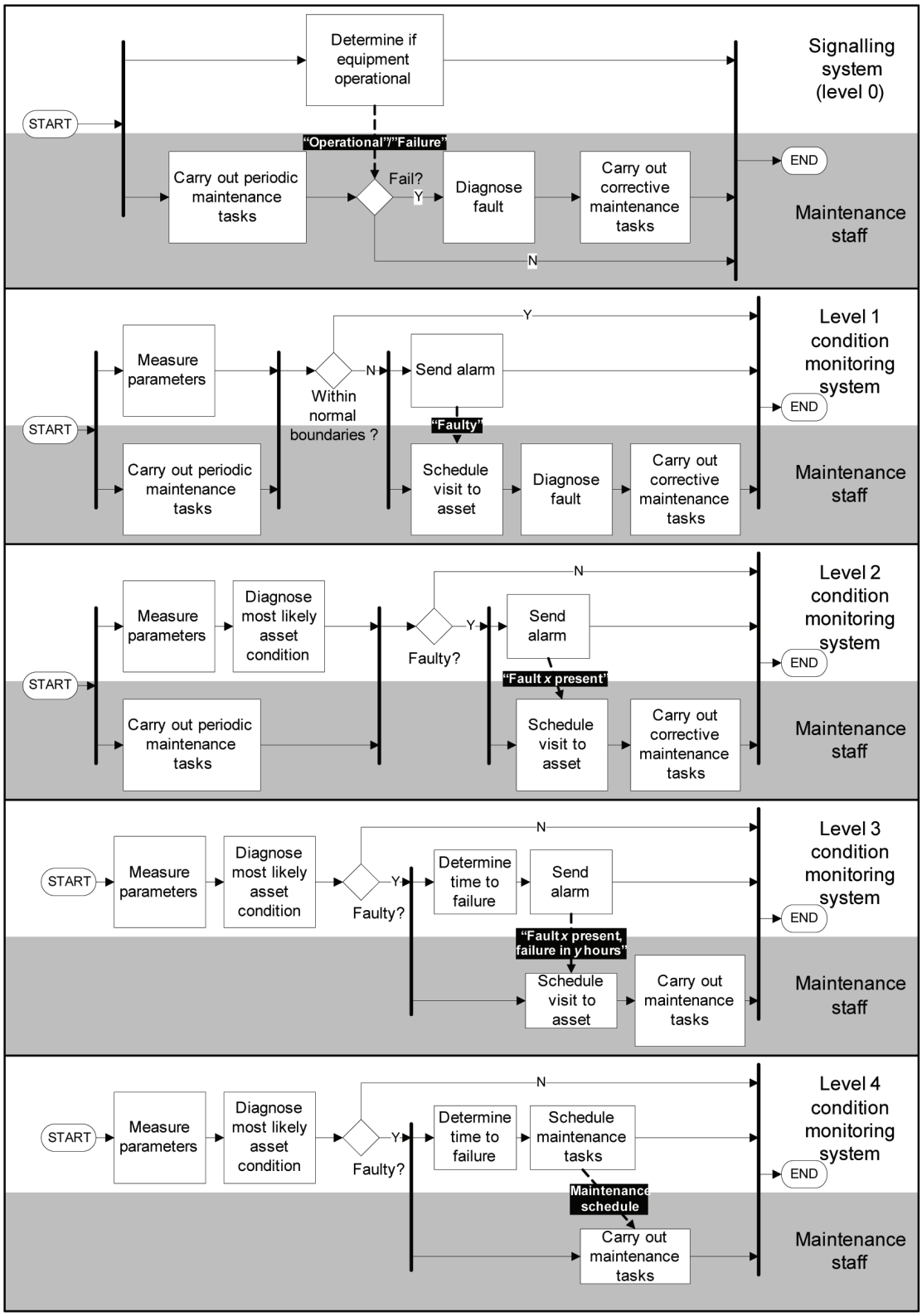


Table 1. Capabilities of a condition monitoring system from Figure 1

Capability Level	Description
0	Failure detection
1	Fault detection
2	Fault detection and diagnosis
3	Fault detection, diagnosis and time-to-failure calculation
4	Automatic maintenance scheduling

The nature of railways means that equipment failures can very often lead to a system halt whilst remedial action is carried out. Train door failures cause delays because of the safety procedures that are followed. For example, certain types of rail vehicle cannot be used if a single door is out of action, because they are passenger emergency exits and must always be available. Failed switch actuators cannot be used to throw points, therefore they limit the control a signaller has over the routing of trains.

However, railway assets are typically collections of relatively low-cost components which are distributed over a large geographic area. Taken in isolation, each part does not justify a large individual effort in condition monitoring, because such effort would cost much more than the asset's value. The challenge is therefore to identify methodologies that can address fault detection, diagnosis and identification in *classes* of asset, resulting in a single strategy which can be applied to a large number of similar assets at once. This achieves better value for money than if detailed methodologies were applied individually for each asset.

Figure 2 shows a categorisation of methods for fault diagnosis. The diagram has been reproduced from a review paper (Venkatasubramanian, Rengaswamy, Yin & Kavuri, 2003), with the addition of automata to the class of qualitative causal models. There are three main categories of fault diagnosis methods: qualitative model-based, quantitative model-based, and process history-based.

2.2.1 Quantitative Model-Based Diagnosis Methods

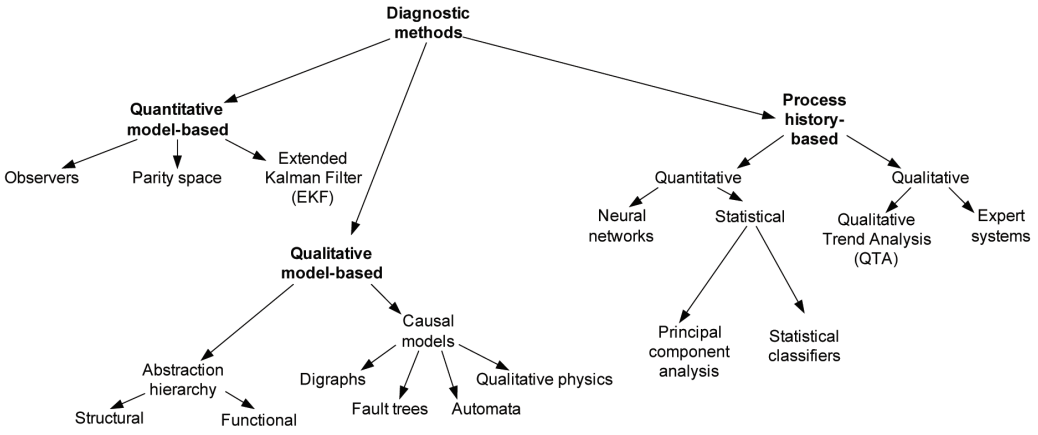
The basic concept of these methods for fault diagnosis is to establish a mathematical model of the plant to be diagnosed, and then predict the values of the plant's measurable variables. Signals known as residuals are generated by comparing the predicted variables to those measured from the plant itself. These residuals can then be used to determine if any faults are present, and what they might be. The most common functions for residual generation are subtraction of the model signal from the measured signal (to detect additive faults) and division of the measured signal by the model signal (to detect multiplicative faults).

Many methods exist for modelling the performance of dynamic systems, but the dominant form of model in the field of fault diagnosis is the state-space input-output model (Venkatasubramanian et al., 2003a) usually in discrete form because variables are measured on a sampled basis.

One advantage of such a model is that it quite often exists already in the controller of a large piece of plant. State-space models are used to estimate the state variables of a system when they cannot be readily measured directly. This model is known as an *observer* and is used to provide more insight into a system's operation and therefore enable it to be better controlled. Mathematical models of this type have been used for some time for various applications including servo motors (Abidin, Rubiyah, Marzuki & Shamsuddin, 2002) and jet engines (Patton & Chen, 1992; Gayme, Menon, Ball, Mukavetz & Nwadiogbu, 2003).

For railway assets, quantitative modelling can be impractical because assets of the same type often have markedly different performance when installed under different circumstances. Without modelling of all possible variables affecting performance, a model-based approach will have

Figure 2. Taxonomy of fault diagnosis methods



inaccuracies which may or may not be acceptable. In all cases, quantitative modelling requires a significant amount of analytical effort in order to effectively create the model and establish the rules for fault diagnosis.

2.2.2 Qualitative Model-Based Methods

There are three main reasons cited in published literature for pursuing the use of a qualitative model. The first is that modelling inaccuracies are not a problem because qualitative methods do not rely on exact numerical modelling. The second is that qualitative methods can be used where it is not possible to make quantitative observations. The third is that qualitative models make it possible to incorporate empirical knowledge about a system in the operational model (Travé-Massuyès & Milne, 1997).

There has been particular focus on the use of non-deterministic (or stochastic) automata as a modelling medium for physical processes (Lunze, 1992). Automata are also known as state machines. Stochastic automata were applied to the observation of qualitative states (Lichtenberg & Lunze, 1997) and the diagnosis of transient faults (Schiller, Schröder, & Lunze, 2001). A study was also carried out into the conditions which must exist for a deterministic automaton model to be valid (Lunze, Nixdorf & Schröder, 1999), and the use of a semi-Markov process model, as a timed description of quantised event sequences, was evaluated (Lunze, 2002). State machines are also used as a qualitative model by (Ramkumar, Philips, Presig, Ho & Lim, 1998).

These papers all deal with situations where variables cannot be directly measured and so a qualitative model must be used to process the qualitative variables which can be measured. The methods involve a partition or quantisation of the measurement space so that a discrete-event representation can be constructed.

Various probabilistic methods then model the transition between states and produce a diagnosis output. A probability level is calculated for each possible fault, which is a useful output because it allows a human observer to see the increasing probability that a certain fault is present over a period of time, and also can show how one fault initially appears more likely, but then diminishes as time goes on.

These methods are effective but there is always a compromise when quantised data is used as an input to a model. The probabilistic diagnosis algorithms are in place to deal with the ambiguity introduced by the use of these data. Quantisation

also means that small changes in measured variables are not always detected straight away. This means that incipient faults may not be detected quickly enough.

Discrete-event models are, however, a very effective means of representing logic systems such as railway signal interlockings, which are highly complex safety-critical systems. Monitoring of such systems could be carried out using discrete-event models to predict the correct sequences of states and outputs.

2.2.3 Process History-Based Methods

History-based methods are fundamentally different from model-based diagnosis methods because no prior knowledge about the monitored system is required. Previously measured data from the system is analysed to produce some knowledge from it which can be used to diagnose faults. This function is known as feature extraction (Venkatasubramanian, Rengaswamy, Kavuri & Yin, 2003). Methods for the application of the knowledge are much the same as those for model-based methods.

There are several categories of feature extraction methods. Quantitative feature extraction can be statistical or non-statistical. Statistical methods are concerned with finding patterns in abstract mathematical representations of data. Principal Component Analysis (PCA) is one of the most recent of these representations and has been applied to face recognition (Kim, Jung & Kim, 2002) and non-destructive testing of metal objects for sub-surface cracks (Sophian, Tian, Taylor & Rudlin, 2003). PCA is a mathematical transformation of vector or matrix data which reduces the dimensionality of the data.

Wavelet transforms are mathematical representations of signals with components in both the frequency and time domains. They have been used for feature extraction in many different fields including fault prediction for ball bearings (Mori, Kasashima, Yoshioka & Ueno, 1996), classification of geographical data (Sveinsson, Ulfarsson & Benediktsson, 2001), classification of NMR spectra (Li, Pedrycz & Pizzi, 2005), wear estimation for industrial turning processes (Pittner & Kamarthi, 1999) and vibration monitoring (Yen & Lin, 2000).

Wavelet transforms are a function of a signal, two scaling variables a and b, and a "mother wavelet" which is a mathematical function with zero mean and finite length. The scaling variables allow the wavelet transform to be carried out at various scales, which can produce a multi-resolution analysis of the signal.

Frequency-domain analysis has been used for some years to extract the features of signals for diagnosis. Fourier transforms in particular can be used to track components of certain frequencies (Lebhrasab, Dassanayake, Roberts, Fararooy & Goodman, 2002).

All these statistical methods rely on complex mathematical functions and produce fairly abstract data which requires further interpretation before it can be of use.

Artificial neural networks (ANNs) are very popular computing methods which have applications throughout engineering. Their basic operation is modelled on that of neurons in the human body, which have input and output components and a vast amount of interconnection. They are capable of learning from input data and performing functions such as modelling of data profiles (which can be used to replace conventional mathematical models) and the classification of patterns (which can be the diagnosis of faults from an initial set of observed residuals generated by a model of any kind) (Basheer & Hajmeer, 2000).

Because of their versatility, ANNs of many different types have been employed in fault diagnosis research in recent times. A set of training data is all that is needed to program them to model systems and diagnose faults, and they have had considerable success. Some examples of previous applications are power transformers (Huang, 2003) and robotic arm manipulators (Vemuri, Polykarpou & Diakourtis, 1998).

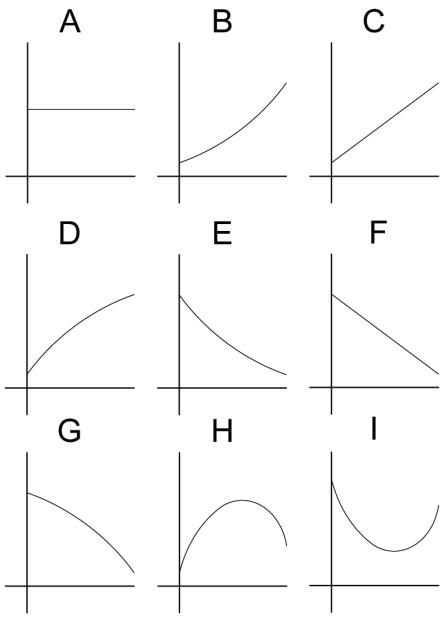
ANNs have the advantage, from a practical point of view, of requiring no detailed analysis of the monitored systems, only a set of training data (although detailed analysis can be incorporated in their design to aid in the learning process). However, their very nature is "black-box"', that is it is difficult to observe how a neural network arrives at a particular decision. It is very important that the decisions made by a fault diagnosis system can be traced right back to their origins so that technician users can interact fully with the system and draw their own conclusions from its output. This is possible with ANNs, but it may not be easy to logically trace the symptoms of a fault in the measured data to the diagnosis produced, so they may not be the best choice as an overall approach for a railway fault diagnosis system. However, they could be used for distinct functions within a system, such as the correct classification of shapes in a Qualitative Trend Analysis system.

Qualitative process history-based methods attempt to solve the problem in non-numerical ways. Two diverse types of qualitative method based on process history are Qualitative Trend Analysis (QTA) and expert systems.

Qualitative Trend Analysis describes the process of qualitatively extracting relevant information from observations and using it to draw conclusions about the state of a monitored system. Cheung & Stephanopoulos (1990) developed a comprehensive scheme for the representation of trends in sensor data. The basic method was to filter the waveform to the scale required for analysis, which removed some of the higher frequency information, and to then represent the waveform as a series of standardised curves.

This enabled sensor data to be partitioned into a sequence of *episodes*, where each episode is a trend of a particular shape. The episodes are represented by an "alphabet" of curve shapes as shown in Figure 3. The problem with the scheme was that the representation of the data constituted a classification problem, for which the authors had no ready solution but referred to the use of neural networks and other complex methods of classification.

Figure 3. 'Alphabet' for qualitative classification of trends



This appeared to make the process very complex, but the idea of using an alphabet of basic shapes to represent waveform data is one which appeals greatly. Some quantitative information can be stored along with the sequence of shape characters. The starting and ending values in each episode are one example of this. Therefore, these representations contain both qualitative and quantitative information, making the approach ideal for this application. If data were collected from many actuators of the same type and in the same condition, it should be possible to represent the data in such a way that the qualitative performance would be identical in all cases (Silmon & Roberts, 2006). This can be achieved by using QTA.

A variation on QTA was proposed, where fuzzy rules were established for particular episodes of behaviour identified in measured data. This approach was shown to be very successful in detecting and diagnosing faults in chemical processes. Given that it is possible to find a set of episodes

in a qualitative representation which are common to all fault conditions for a given actuator, fuzzy rules established for particular episodes (those where the effects of faults are the greatest) could be used to great effect. The fuzzy sets developed in this paper were of uniform size, translating to heuristic categories such as "average", "large", or "very small" (Ozyurt, Hall & Sunol, 1999).

Expert systems are methodical representations of human knowledge which diagnose faults by reasoning with IF-THEN rules. This can be done manually or using one of several expert system building tools. Their chief advantages are the ease with which they can be developed and the transparency of their reasoning. Their chief drawbacks are that they are very specific to particular systems and are difficult to update.

The power distribution industry has focussed heavily on the use of expert systems for the diagnosis of faults in components such as transformers and insulators. Dissolved gas analysis from transformer oil has been used to establish a set of diagnosis rules for the detection of faults in transformers (Lin, Ling & Huang, 2003). This was found to be very successful, with detection rates for the tested faults in excess of 90%.

Butler (1996) combined an expert system engine, built using the EXSYS automated tool, with a neural network model of the aging of power components to form a diagnosis system for large networks of power distribution equipment. This was then used to find the location, within the network, of equipment which was behaving abnormally due to aging. Neural networks and expert systems were also combined for the diagnosis of steady-state faults in chemical processes (Becraft & Lee, 1993), and for the diagnosis of power transformers (Wang, Liu & Griffin, 1998).

2.3 Fault Diagnosis on the Railways

Traditionally, the diagnosis of faults in railway equipment has been carried out, once a failure has occurred, by technicians called to the equipment's location. According to failure statistics from the Southern region of the British network, between 17% and 22% of failures on point actuators could not be traced to a particular fault when the technicians examined them, resulting in a record stating "Tested OK on arrival", despite the fact that the actuator had failed in service and therefore a fault was clearly present (Advantage Technical Consulting, 2002).

Clearly, a human examination is not sufficient to detect all the possible faults in these actuators. More insight is required into the operation of the actuator, and so automated condition monitoring has emerged as a potential solution. Commercial solutions focus, in the main, on diagnosing abrupt faults using large arrays of sensors to detect faults in the positioning and locking mechanisms of the actuator being monitored (Zhou, Duta & Henry, 2002). There are currently several pilot schemes operating around the UK.

Little academic research has been carried out to advance this technology, however a number of papers have been published which aim to gain a better insight into the dynamics of an actuator and thereby to try and detect incipient faults which would not be seen on standard condition monitoring equipment until a failure was absolutely imminent.

A neuro-fuzzy diagnosis approach has been proposed for the diagnosis of pneumatic point machines, where the variables measured were first partitioned into a small number of regions (Roberts, Dassanayake, Lehrasab & Goodman, 2002). This allowed a piecewise linear input-output model to be used. Partitioning the operational variables in the same way, it was possible to generate residuals which were sensitive to faults. Local nodes on each actuator were capable of distinguishing sensor faults from actuator faults. Measured data from actuator faults was forwarded over a network to a central processor which ran an Adaptive Neuro-Fuzzy Inference System (ANFIS) to diagnose the fault.

This system is interesting because it uses a simplified model in order to reduce complexity. However, it does not fully address the issue of tuning the rule base so that the rules apply equally to each instance of an actuator type. In practice, this system would have required the faults to be simulated on each actuator in order to correctly tune the model, something which would not be possible on the railways.

Neural networks have been used to model the performance of pneumatic train doors and to diagnose faults (Lehrasab et al., 2002). An RBF neural network modelled the displacement profile of the door to a level of accuracy which could be predefined. Complexity of the network was reduced by filtering and removing corrupt values from the data set. Diagnosis was carried out with a Self-Organising Feature Map (SOFM) neural network. The results published in this paper indicated that the accuracy of the system was roughly 80%.

2.4 Conclusions from Literature

Both qualitative and quantitative fault diagnosis methods, be they model-based or not, have valuable properties. However, the problem here is to detect, diagnose and identify faults in a large set of assets of the same type, where the qualitative behaviour of each is similar, but the quantitative behaviour is not. A new approach is required because neither a purely qualitative nor a purely quantitative approach is likely to succeed.

There are many problems with the use of quantitative models. The most obvious is that real world equipment can never be modelled with 100% accuracy. Model inaccuracies and uncertainties mean that residuals are unlikely to be solely dependent on faults, and this can lead to spurious detection of faults. Quantitative models are only suitable where sufficient detail can be obtained about an asset's behaviour to ensure a highly accurate model.

Qualitative modelling is of particular interest because it has the potential to solve the problem of quantitative variation between instances of similar actuators. It is fair to assume that all instances of a particular actuator will share similar characteristics of performance when viewed qualitatively.

However, the results of the research carried out into qualitative modelling suggest that the ambiguity introduced may impair the ability of a diagnosis system to detect incipient faults at an early stage of detection. In many cases, qualitative models are no simpler or easier to understand than quantitative ones. The effectiveness of discrete-event modelling means that it is very suitable for the diagnosis of non-incipient faults in logical systems such as signalling, but less suited to the subtle deterioration of incipient faults in mechanical equipment.

Model-based methods in general may well be suited to applications where the expense of the modelling task is balanced by the size or value of the installation, as may be the case with electrical substations or signal interlockings. In the latter case, a qualitative model may be extracted without difficulty from the logical equations used to design the system. For distributed, low-value assets, model-based methods may be uneconomical unless the model is equally applicable to the entire set of assets and does not need excessive extra analysis each time it is deployed on a new instance.

Process history-based methods are suited to problems where sufficient modelling detail is not available to create an accurate model, either qualitative or quantitative. These methods require subtle and sophisticated algorithms to handle the data in a way which is sensitive to the context of the asset and its behaviour. For assets such as track circuits, switch actuators, train doors and level crossing barriers, a history-based method allows a small amount of analysis to propagate the benefits of condition monitoring to unlimited asset instances.

3. CONDITION MONITORING CASE STUDY

3.1 HW Electro-Mechanical Switch Actuator

Switch actuators are used to change the direction of trains at junctions by moving, and locking in place, movable rails known as "switch rails". The switch rails must be locked in place before trains can be signalled over the switch; this is usually verified using finely calibrated detection mechanisms, known as end position detectors, connecting the switch rails to relay contacts. Small changes in ambient conditions, lack of lubrication, obstructions or the adjustment of the switch mechanism can lead to a loss of end position detection, resulting in disruption to train services.

3.1.1 The Actuator in Detail

The HW switch actuator is in widespread use on the railways of the UK. It is an electro-mechanical actuator using a DC permanent magnet motor, clutch, reduction gears and reciprocating linkage to exert a horizontal force on the ends of the switch rails. The single drive rod moves both switch rails by pushing on a fixed-length stretcher bar which is bolted to both rails. The connection between the stretcher bar and the actuator drive is adjustable, which allows variation in the locking forces at both end positions.

3.1.2 Fault Simulations

The focus in this case study was on the detection of incipient adjustment faults in the switch mechanism. To simulate these, the drive link or backdrive was gradually adjusted to provide too much or too little force at the locking point of the switch's movement. Excessive adjustment resulted in failure of the switch to complete its movement, so adjustments were small and incremental. The following faults were simulated:

- Too much leverage in the backdrive
- Too little leverage in the backdrive
- Drive link adjusted to provide excessive locking force on the normal side
- Drive link adjusted to provide excessive locking force on the reverse side

3.1.3 Key Parameter Measurements

The key parameters which can be affected by faults in this actuator were identified as motor current, displacement of the switch and the horizontal force in the drive (Silmon, 2009).

Current was measured using a split-core current transducer attached to the motor wires such that throws in either direction registered a positive current. A drawstring displacement transducer, based on a potentiometer, was attached to the drive arm and the body of the actuator, allowing displacements of up to 250 mm to be measured. The throw of a switch actuator is typically 150-200 mm. The force in the drive was measured using a specially-made load pin which fitted in the drive link in the place of a bolt. The data collected is shown in Figure 5.

3.1.4 Analysis Algorithm

The data were analysed using a Qualitative Trend Analysis approach. In the training phase, the algorithm identified key shapes in the measured waveforms and created fuzzy diagnosis rules based on the changes in those shapes under severe fault conditions. When "deployed" (i.e. when presented with data from a simulation of operation), the algorithm identified the same shapes and looked for the changes in those shapes which indicated particular faults. The result was an increase in the fuzzy membership values for the fault which was

Figure 4. Flow diagram showing the general process of the QTA-based fault detection algorithm

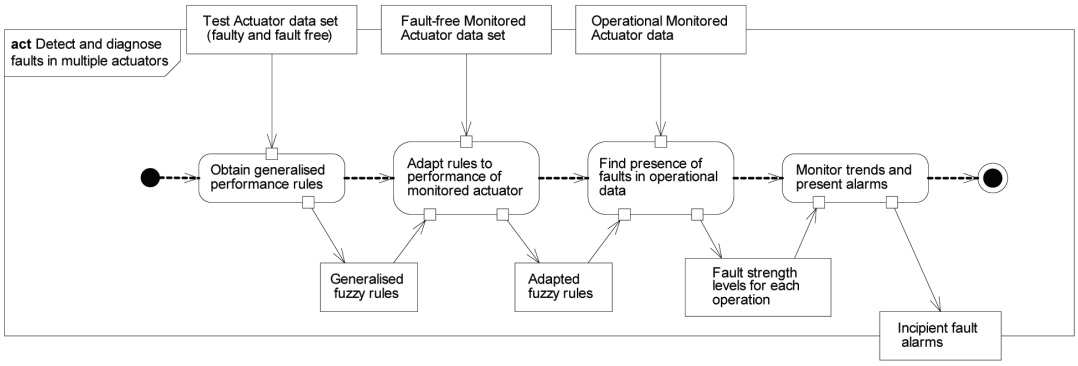
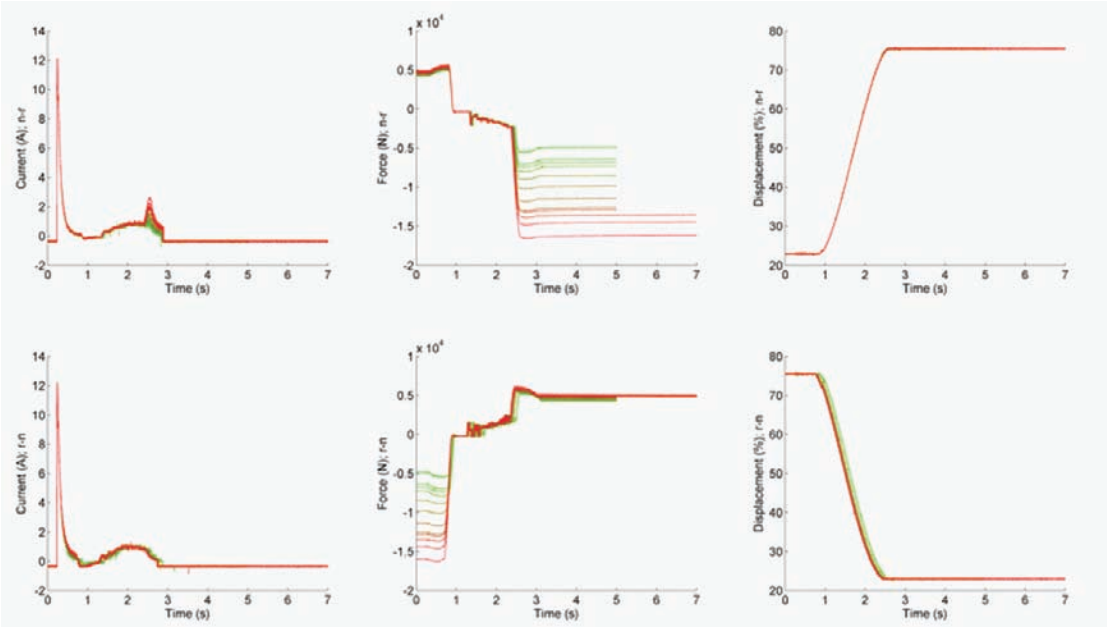


Figure 5. Graphs of current, force and displacement for the HW switch actuator with progressive over-driving of the backdrive (n-r = normal to reverse, r-n = reverse to normal, green waveforms are less faulty, progressing to red as severity increases)



being simulated. An overview of this process is illustrated in Figure 4.

Because the rules were based on changes or differences rather than absolute values, they were equally valid on actuators other than the one used to train the system. The sequence of shapes in the waveform was used to identify the correct values to evaluate against the rules.

3.1.5 Testing and Results

The algorithm was tested on a group of three actuators. The same data were taken from each, allowing testing in a number of combinations, where one actuator was used to form the rules, and another was used to test them, simulating the real-life situation where training would be carried

Table 2. Results of testing of the algorithm, showing the numbers of combinations which were successful

Test spec	Fault	Normal-reverse		Reverse-normal	
		Combinations	% success	Combinations	% success
1	Fault free	6 of 6	100	6 of 6	100
	Overdriving to normal	6 of 6	100	6 of 6	100
	Overdriving to reverse	6 of 6	100	5 of 6	83
	Backdrive overdriving	1 of 2	50	2 of 2	100
	Backdrive underdriving	1 of 2	50	2 of 2	100
2	Fault free	6 of 6	100	6 of 6	100
	Overdriving to normal	6 of 6	100	6 of 6	100
	Overdriving to reverse	5 of 6	83	4 of 6	67
	Backdrive overdriving	1 of 2	50	2 of 2	100
	Backdrive underdriving	0 of 2	0	0 of 2	0

out on an actuator in a lab or training school and the rules deployed on all monitored actuators in the asset base.

The first test specification was that the system should express a positive membership trend for the fault being simulated, signifying correct fault detection. The second was that the gradient of the membership trend for the simulated fault should be greater than that of other trends, signifying correct diagnosis of the fault (see Table 2).

4. CHALLENGES FOR THE FUTURE

Here we will outline the research areas for the future, and the challenges identified through experience in implementation of advanced techniques in the railway sector. These include:

- The need for portable and transferable information across organisational boundaries, including an examination of the work of the InteGRail project and its future extension
- Designing condition monitoring systems to interact effectively with human operators in order to achieve overall benefits and reliability improvements
- Improvements to the accuracy of models
- Limitations of automatic condition monitoring and possible methods for overcoming them

REFERENCES

Abidin, M. S. Z., Rubiyah, Y., Marzuki, K., & Shamsuddin, M. A. (2002). Application of a model-based fault detection and diagnosis using parameter estimation and fuzzy inference to a DC servomotor. *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, (pp. 783-788).

Advantage Technical Consulting. (2002). *Review of the reliability of point motors and track circuits*.

Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: Fundamentals, computing, design and application. *Journal of Microbiological Methods*, *43*, 3–31. doi:10.1016/S0167-7012(00)00201-3

Becraft, W. R., & Lee, P. L. (1993). An integrated neural network/expert system approach for fault diagnosis. *Computers & Chemical Engineering*, *17*(10), 1001–1014. doi:10.1016/0098-1354(93)80081-W

Butler, K. L. (1996). An expert system-based framework for an incipient failure detection and predictive maintenance system. *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems*, (pp. 321-326).

Cheung, J. T.-Y., & Stephanopoulos, G. (1990). Representation of process trends, part I: A formal representation framework. *Computers & Chemical Engineering*, *14*(4-5), 495–510. doi:10.1016/0098-1354(90)87023-I

Gayme, D., Menon, S., Ball, C., Mukavetz, D., & Nwadiogbu, E. (2003). Fault detection and diagnosis in turbine engines using fuzzy logic. *NAFIPS 2003: 22nd International Conference of the North American Fuzzy Information Processing Society*, (pp. 341-346).

Huang, Y. C. (2003). Condition assessment of power transformers using genetic-based neural networks. *IEE Proceedings. Science Measurement and Technology*, *150*(1), 19–24. doi:10.1049/ip-smt:20020638

Kim, K. I., Jung, K., & Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, *9*(2), 40–42. doi:10.1109/97.991133

Lehrasab, N., Dassanayake, H. P. B., Roberts, C., Fararooy, S., & Goodman, C. J. (2002). Industrial fault diagnosis: Pneumatic train door case study. *Proceedings of the IMechE, Part F: Rail and Rapid Transit, 215*(11), 27-46.

Li, D., Pedrycz, W., & Pizzi, N. J. (2005). Fuzzy wavelet packet based feature extraction method and its application to biomedical signal classification. *IEEE Transactions on Bio-Medical Engineering*, *52*(6), 1132–1139. doi:10.1109/TBME.2005.848377

Lichtenberg, G., & Lunze, J. (1997). Observation of qualitative states by means of a qualitative model. *International Journal of Control*, *66*(6), 885–903. doi:10.1080/002071797224441

Lin, C. E., Ling, J. M., & Huang, C. L. (2003). An expert system for transformer fault diagnosis using dissolved gas analysis. *IEEE Transactions on Power Delivery*, *8*(1), 231–238. doi:10.1109/61.180341

Lunze, J. (1992). Qualitative modelling of continuous-variable systems by means of non-deterministic automata. *Intelligent Systems Engineering*, *1*(1), 22–30. doi:10.1049/ise.1992.0003

Lunze, J. (2000). Diagnosis of quantized systems based on a timed discrete-event model. *IEEE Transactions on Systems, Man, and Cybernetics*, *30*(3).

Lunze, J., Nixdorf, B., & Schröder, J. (1999). Deterministic discrete-event representations of linear continuous-variable systems. *Automatica*, *35*, 395–406. doi:10.1016/S0005-1098(98)00176-9

Mori, K., Kasashima, N., Yoshioka, T., & Ueno, Y. (1996). Prediction of spalling on a ball bearing by applying the discrete wavelet trasform to vibration signals. *Wear*, *195*, 162–168. doi:10.1016/0043-1648(95)06817-1

Özyurt, I. B., Hall, L. O., & Sunol, A. K. (1999). SQFDiag: Semiquantitative model-based fault monitoring and diagnosis via episodic fuzzy rules. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *29*(3), 294–306. doi:10.1109/3468.759283

Patton, R., & Chen, J. (1992). A robustness study of model-based fault diagnosis for jet engine systems. *Proceedings of 1st IEEE Conference on Control Applications,* (pp. 871-876).

Pittner, S., & Kamarthi, S. V. (1999). Feature extraction from wavelet coefficients for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(1), 83–88. doi:10.1109/34.745739

Ramkumar, K. B., Philips, P., Presig, H. A., Ho, W. K., & Lim, K. W. (1998). Structured fault-detection and diagnosis using finite state-automaton. *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, (pp. 1667-1672).

Roberts, C., Dassanayake, H. P. B., Lehrasab, N., & Goodman, C. J. (2002). Distributed quantitative and qualitative fault diagnosis: Railway junction case study. *Control Engineering Practice*, *10*(4), 419–429. doi:10.1016/S0967-0661(01)00159-9

Schiller, F., Schröder, J., & Lunze, J. (2001). Diagnosis of transient faults in quantised systems. *Engineering Applications of Artificial Intelligence*, *14*, 519–536. doi:10.1016/S0952-1976(01)00020-3

Silmon, J. A. (2009). *Quantification of benefit available from switch and crossing monitoring.* INNOTRACK Research Project Deliverable. Retrieved from www.innotrack.eu

Silmon, J. A., & Roberts, C. (2006). A systems approach to fault detection and diagnosis for condition-based maintenance. *Proceedings of the 1st IET International Conference on Railway Condition Monitoring*, 2006.

Sophian, A., Tian, G. Y., Taylor, D., & Rudlin, J. (2003). A feature extraction technique based on Principal Component Analysis for pulsed eddy current NDT. *NDT & E International*, *36*, 37–41. doi:10.1016/S0963-8695(02)00069-5

Sveinsson, J. R., Ulfarsson, M. O., & Benediktsson, J. A. (2001). Cluster-based feature extraction and data fusion in the wavelet domain. *IEEE International Geoscience and Remote Sensing Symposium*, (pp. 867-869).

Travé-Massuyès, L., & Milne, R. (1997). Gas-turbine condition monitoring using qualitative model-based diagnostics. *IEEE Expert*, 22–31. doi:10.1109/64.590070

Vemuri, A. T., Polykarpou, M. M., & Diakourtis, S. A. (1998). Neural network based fault detection in robotic manipulators. *IEEE Transactions on Robotics and Automation*, *14*(2). doi:10.1109/70.681254

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., & Yin, K. (2003). A review of process fault detection and diagnosis, Part III: Process history based methods. *Computers & Chemical Engineering*, *27*, 327–346. doi:10.1016/S0098-1354(02)00162-X

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003a). A review of process fault detection and diagnosis, Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, *27*, 293–311. doi:10.1016/S0098-1354(02)00160-6

Wang, Z., Liu, Y., & Griffin, P. J. (1998). A combined ANN and expert system tool for transformer fault diagnosis. *IEEE Transactions on Power Delivery*, *13*(4), 1224–1229. doi:10.1109/61.714488

Yen, G. G., & Lin, K. C. (2000). Wavelet packet feature extraction for vibration monitoring. *IEEE Transactions on Industrial Electronics*, *47*(3), 650–667. doi:10.1109/41.847906

Zhou, F. B., Duta, M. D., & Henry, M. P. (2002). Remote condition monitoring for railway point machine. *Proceedings of the ASME/IEEE Joint Rail Conference*, (pp. 103-108).

Chapter 17
Security of Railway Infrastructures

A. Di Febbraro
University of Genoa, Italy

F. Papa
University of Genoa, Italy

N. Sacco
University of Genoa, Italy

ABSTRACT

In recent years, some sadly famous terrorist attacks that occurred in different countries have put into evidence that railway transportation systems are not suitably protected, and not capable of tolerating and promptly reacting to them.

Moreover, it is clear that such mass transportation systems are particularly attractive for terrorists, due to the potentially far-reaching, often "spectacular" results of attacks. Examples of such kinds of events are the New York (2001), Madrid (2004), and London (2005) terrorist attacks. In addition, by focusing on ground transportation networks, and especially on railway systems, it is also easy to observe that they are particularly difficult to be secured since they are characterized by high accessibility and wide extension, as also noted by Fink (2003). In this sense, the needs of security and of mobility often conflict with each other. In effect, while an open and accessible system provides an efficient transportation of people and goods, this openness also allows malicious entities to exploit the transportation system as a target, weapon, or means to reach another target (Murray-Tuite, 2007). Then, on the contrary, it is clearly evident that security actions taken to limit malicious adversaries from reaching or capturing their targets may degrade the transportation system performances, so they have to be designed with particular attention. This is the reason why worldwide institutions are more and more sensitive to the growing need for security of the so-called Critical Infrastructures (CI), such as railway transportation systems, and are adopting a number of regulatory measures (US Congress, 2007; EU Commission, 2005, 2008, and 2010).

DOI: 10.4018/978-1-4666-1643-1.ch017

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

For what concerns scientific research, the efforts are intended to define methodologies, build risk mitigation devices, and find out best practices that are technologically advanced, soon achievable, reliable, so as to increase the infrastructure protection without affecting the relevant transportation system performances. In this framework, Quantitative Risk Analysis (QRA) represents the main methodological approach for assessing security, which is indeed often characterized by a large set of variables dependent on human sensitivity, and requires calibration and adaptive tuning, thus resulting into unfriendly tools for the non-skilled users.

Then, in this chapter, to tackle with the problem of clarifying the aims, the characteristics, and the limitations, a general architecture for a possible QRA tool for railway security assessment is presented, with particular attention to the relevant specifications (Di Febbraro et al., 2010).

The chapter is organized as follows: In section 1, the basic definitions of the security risk analysis and the characteristics of the railway security problem are introduced, and a bibliography review is reported. Then, in section 2, the general architecture for designing a security risk analysis tool is presented, focusing on the relevant specifications, and on the input/output characteristics. Therefore, in section 3, with the aim of pointing out the characteristics of the presented architecture, an explicative case study is defined based on real world data coming from Italian railways. Finally, some conclusions and remarks are discussed in chapter 4.

DEFINITIONS, MOTIVATIONS AND BIBLIOGRAPHY REVIEW ON RAILWAY SECURITY

In this section a basic glossary of the terms used in the chapter is provided, and the basic characteristics that make railway systems of significant interest for security are discussed. In the end, a bibliography review is presented.

Basic Glossary

In this section, a brief glossary of the most common terms used in the chapter is provided with the aim of facilitating the comprehension of the following sections. Then, consider the following basic definitions:

- **Threat:** The potential intent to cause harm or damage to properties or people;
- **Attack Likelihood:** An estimate of the real probability/frequency of a real attack
- **Terrorism:** A deliberate use of violence against people or properties with the aim of intimidating or coercing a government, the civilian population in furtherance of political or social objectives;
- **Sabotage:** A deliberate action aimed at weakening an enemy through subversion, obstruction, disruption, and/or destruction. Unlike terrorists, saboteurs do not consider fatalities the primary objective, although they do not exclude them;
- **Robbery/Theft:** The use of force or violence against properties or people with the aim of depriving the rightful owner of property;
- **Vandalism:** The use of force or violence against property with intent of malicious destruction of defacement of public or private property;
- **Adversary:** A general term indicating terrorists, saboteurs, thieves and vandals;
- **Attractiveness:** A measure of the likelihood of an attack to an asset;

- **Vulnerability:** A measure of the easiness of a security protection system to be overcome by adversaries.

The interested reader may refer to ASIS International (2003) and Garcia M. L. (2001) for a more complete glossary.

Railway Security Threats

In this section, the main threats for railway security are. In doing so, it is worth saying that three different kinds of threats are usually considered: terrorism/sabotage (gathered since they often use the same tactics, equipment, and so on, although with different aims), theft, and vandalism. The last two are the more frequent attacks so that, although they normally cause relatively low damages, often represent the main threats railway systems must face.

Anyway, since characterizing the threats for any target is a difficult task to tackle with, in this section some considerations about the characteristics making threats up are discussed. In this framework, it is worth saying that adversaries presumably *assess* potential targets by taking into account the relevant accessibility, susceptibility to damages or injuries, and how much the accomplishment of a malevolent act matches their goals. In other words, a significant number of factors influence the decision of an adversary to attack a target or not. These factors may be pointed out by the following considerations:

- **Knowledge of the Existence of Potential Targets:** An adversary must know that a target exists before considering it. The more widespread the knowledge of the target's existence, the larger the pool of potential adversaries. It is evident that railway transportation systems cannot be hidden;
- **Availability of Information About Potential Targets and Relevant Protections:** Adversaries tends to choose targets about which they can gather the most information, while a lack of information can lead to doubts and make a target relatively less attractive than one for which information is readily available. This is the reason why the "openness" of railway systems makes them attractive;
- **Symbolic or Ideological Relevance of Potential Targets, and Publicity Generated by a Successful or an Attempted Attack:** Terrorists often focus on such targets with high symbolism. In this sense, even if the objective is not to cause extensive damage or to kill and injure many, may constitute targets for terrorists and saboteurs. In terms of railway systems, new infrastructures or the most technological advanced ones are reasonably more attractive than the others;
- Other factors to be take into account are *the potential mass casualty*, *the perceived criticality of systems*, or *the potential economic disruption* which constitute factors that greatly influence attractiveness of an asset;
- The last factor that significantly influences the decision of vandals and thieves is the *required level of effort:* in this sense, a target requiring a high effort has lower attack likelihood than one requiring a high effort, both in terms of equipment and number of adversaries.

It is evident that the above factors that greatly influence the attack likelihood are of qualitative nature and usually not measurable. In addition, they are often provided by heuristic considerations that evidently depend on the knowledge and sensitivity of the security experts that asses risk and design the protection systems. This is the reason why, in

Table 1. Transportation related terrorism events from 2000 to 2004 by transportation mode (based on Global Terrorism Database data)

Year	Air Transportation	Maritime Transportation	Ground (railway and road transportation)
2004	2	5	25
2003	1	5	41
2002	0	10	41
2001	2	12	28
2000	2	11	56
mean	*1.4 per year*	*8.6 per year*	*38 per year*

the following sections, the problems arising from the non-measurability and on the low reliability of such factors will be addressed and discussed.

Then, before introducing a general architecture for railway security risk analysis tools, in the following sections some considerations about the different "components" of railway infrastructures, and the relevant threats or famous past events, are discussed.

Passenger Rail Systems

Passenger rail services may take different forms (heavy rail, commuter rail, underground rail, light rail) but share certain characteristics, also with road transportation, that make them vulnerable to attacks:

- They make scheduled stops along fixed routes;
- Their operate with people having quick and easy access to stations and trains;
- The number of access points, the need of quickness, and the volume of ridership make people screening like in air transportation impossible in practice.

In view of these considerations, it could be stated that surface transportation systems such as railroads and mass transit remain hard to protect because they are easily accessible and extensive. In addition, as said, such systems are characterized by great attractiveness, as shown by the relatively high number of attacks ground transportation have been subjected to, and partially reported in Table 1, where it is possible to note that from 200 to 2004 ground transportation systems have been attacked about 4.5 times more than maritime transportation systems, and about 27 times more than air transportation (Zeng et al., 2007).

For what concerns passenger stations, they are sites of very high attractiveness for terrorists and thieves due to the high concentration of people gathered in. Anyway, stations do not represent a class of special asset found only in railway systems that requires ad-hoc risk analysis methodologies, so they will not be considered in the present chapter. On the contrary, the railway assets here considered can be classified into three general categories: track, signaling system, and power supply assets, yards and equipment. Each element implements different functions, resulting into a unique characteristics, attractiveness and vulnerability, as described in the following.

Tracks and Rails

The railway track infrastructure includes what trains ride on (railways), ride through (tunnels), and ride across (bridges and culverts). In addition, railways are composed of:

- Ballast, which supports crossties and hence rails. Its functions are to restrain the track ties from movements, both under static and dynamic loads, while providing drainage for the track, and to keep crossties and rails at the proper elevation and alignment;
- Crossties that provide support for the rails, communicate, and distribute rail loads to the ballast, thus maintaining rails at the proper gauge and alignment;
- Rails support the wheels of the train, and provide a smooth surface for the wheels to run over. Any train derailment from rails is significantly damaging to equipment and track, and requires considerable time and expense to remedy.

Attacks to the track infrastructure not only require a low level of effort of adversaries, but also provide a great fan of opportunities. Attacks to tunnels, bridges, or culverts require a significantly greater level effort of adversaries, and in general need some sort of explosive or pyrotechnic devices. In addition, with such kind of attacks, the collateral effects may result into significant damages to trains, for instance when debris from a collapsed portion of a tunnel blocks the track.

Signaling Systems

The purpose of the signaling systems is to provide information to trains and to control dispatch centers about the state of the track. They physically consist of a set of computers and electric circuits connected to a control or dispatch centers via wired or wireless communication channels that compute and provide information to the trains by means of colored lights, mounted above or adjacent to the railway, or even inside the trains. Different light configurations have different meanings, and gives different information about the occupancy of the track ahead, the allowed movement and authorized speed, as well as on the position of the switches.

Evidently, such vital systems may be potential targets for adversaries. Anyway, attacks to signaling system are more difficult to execute than attacks on the track/rail infrastructure, and may require computer attacks by means of hacking, or only "physical" attacks to the infrastructures. While these last kind of attacks is, in general, tolerate and "compensated" directly by the signaling systems (normally designed to be fault tolerant), the system hackings require a detailed understanding of the whole system, since the attack must overcome their failsafe design.

Therefore, since in the present chapter only physical attacks are considered, the signaling systems threats will not be developed in detail.

Power Supply Assets, Yards and Equipment Warehouses

Power supply sites, storing yards, as well as locomotive and equipment warehouses, provide several possibilities of attacks. In this framework, the first threat consists of gaining their physical possession of the assets therein, just for theft, or even with the aim of using them as a mean for an attack.

In addition, due to the large number of rail cars and materials gathered in yards, where trains with potential dangerous goods are received and stored until being delivered, these sites may guarantee access to a wide range of hazardous material, and in large quantities (Peterman D.R., 2006). Therefore, the proximity of such sites to densely populated areas means that the loss of control of cars, or the release of their contents may affect a large number of people. In addition, the access of adversaries to rail yards is relatively easy, thus providing the attacker a certain degree of freedom in controlling the material release. In effects, many yards are not fenced, and even when fenced, they may be easily accessed, since railways do not usually employ sophisticated intrusion detection systems. In addition, although larger yards have railroad police conducting regular security sweeps, the

limited number of officers available for this task and the size of the yard reduce the effectiveness of these patrols.

Then, to conclude, it is of paramount importance to face physical attacks to assets of this, not only in terms of the possible consequences that terrorists or saboteurs may cause, but also in term of theft and vandalism, being the assets gathered in yards often of significant economic value.

State of the Art

In this section, a brief literature review is reported with the aim of pointing out the peculiarities of the proposed architecture.

Then, the analyzed literature on security essentially tackles with the following problems:

1. Identify what can go wrong, that is what malicious attacks are possible (Kaplan and Garrick, 1981; Fink, 2003), estimate the relevant likelihood, that is how much they are probable or frequent (Aven, 2007; Kaplan and Garrick,1981; Fink, 2003);
2. Identify and estimate what are consequences of attacks to critical infrastructures (Apostolakis and Lemon, 2005; Di Febbraro and Sacco, 2010);
3. Take the most appropriate decisions about the investments for improving the protection of assets (Garrick et al., 2004; Lambert and Farrington, 2007).

Further readings about risk analysis in both security and safety subjects, about critical infrastructures, and the applications to transportation systems can be found in (Amin, 2002); Eames, 1999; Farrow, 2004; Hartong et al. 2008; Hood et al.,, 2003; Raj and Pritchard, 2000; Rowshan et al., 2005).

To conclude, since the tool described in this chapter explicitly faces the problem of assessing risk when some inputs and outputs are of qualitative nature and/or provided throughout subjective estimations of non-measurable variables (Kaplan, 1992), by means of a fuzzy logic approach, the interested reader can find detailed information about the general characteristics of such a modeling approach and its application to risk assessment and reliability analysis in (Zadeh et al, 1968; Klir and Yuan, 1995) and (Ravi et al., 2000; Bajpai et al., 2009; Akgun et al., 2010), respectively.

Review of the Main Modeling Approaches for Risk Assessment

As above mentioned, in this chapter Fuzzy Logic will be discussed for performing risk assessment. Despite this, it is worth saying that other possible approached to the considered problem are possible, and then, in this section, some information about the different modeling approaches which may results helpful in assessing risk are presented, with the aim of discussing briefly their possible uses and to introduce the relevant references. Then in the following paragraphs, it will be discussed why it results to be useful to consider railway systems and the attack of adversaries as Discrete Event Systems (DESs), also introducing the relevant main modeling approaches.

Then, for what concerns DES, they may be defined as those systems in which the state can assume discrete, even non-numerical or logic, values (Cassandras and Lafortune, 2008). In this kind of systems, an event e_h , $h = 1,\ldots,H$, causes a transition from one state x_k to another state x_{k+1} by means of the state equation $x_{k+1} = \delta\left(x_k, e_k\right)$, $k = 0,1,\ldots$ which may be expressed, in practice, by as a mathematical equation, as a flow chart, or even as a computer simulation program.

The models of this kind of systems do not usually take into account the micro-changes occurring continuously, but simply consider those macro-changes driven by the occurrence of events. This is, in effect, the reason why, in DESs, the time variable depends on the event occurrence, being it updated when, and only when, an event occurs.

Another characteristic that makes DES useful in security risk analysis is their capability of representing sequence, synchronization, and parallelism of events, thus resulting to be suitable for representing sequences of actions performed by attackers, particular conditions that have to be met at a time, and independent systems dynamics, respectively.

As regards the applications to security risk analysis, such classes of systems and models appear to be capable of representing the railway dynamic, as well the sequence of actions that an adversary has to perform during an attacks, the conditions of the railway systems that make an attack feasible and so on.

For what concerns the analytic tools for representing DES, the most important to be mentioned are Petri Nets (Murata, 1989) and Discrete Time Markov Chains. Both such formalisms are able to represents the main dynamics of DES, also providing analytical tools for performance analysis (for instance, for computing the time elapsing from the beginning of an attack to reaching of the target), or structural properties (such as the verification of the conditions to be reached so that an attack becomes possible). Anyway, since discrete event systems are usually too complex to be analyzed analytically, they are often simulated by means of computer models (Banks et al, 2000).

To conclude it is worth mentioning also Fault Tree analysis, Event Tree Analysis, and Bayesian Networks that may result to be useful in security risk assessment. An intersted readred may refer to (Lee et al., 1985), (Hong and Dugan, 2004) and (Jensen, 2001) for detailed description of such formalisms.

A GENERAL ARCHITECTURE FOR RISK ANALYSIS TOOLS

In this section, the proposed risk analysis methodology is presented. To do so, the architecture of a complete risk analysis tool is described and the relevant specifications are discussed.

The section is organized as follow. In the first part, the *spatial discretization* of railway infrastructure is discussed with the aim of determining the set of assets or sites constituting it. Then, the architecture of a general risk analysis tools is described. Finally, the detailed descriptions of the elements constituting the proposed tool architecture are provided.

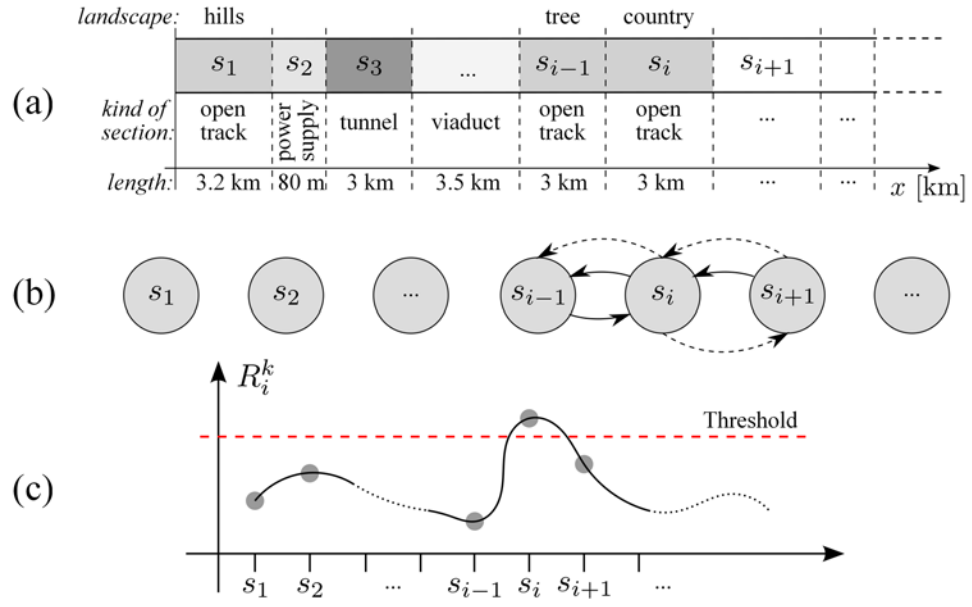
Railway Infrastructure Discretization

As previously said, any railway line consists of a *geographically distributed infrastructure* whose characteristics vary along the line itself due to the presence of different landscapes (which modify the accessibility of sites), or due to the presence of viaducts, galleries, and so on. In addition, alongside railways, there are assets characterized by different attractiveness and/or vulnerability, and hence by different risk level, such as power supply stations, telecommunication centers, storage yards, and so on. Then, let $S = \{s_i\}$ be the set of the georeferenced sites and sections that constitute a generic railway line. In such a set, sites are the railway elements whose longitudinal dimension is negligible with respect of the whole railway (that is, under 100-200 meters), whereas sections are the railway elements whose longitudinal dimension is not negligible (that is, from 200 meters until some kilometers).

In the proposed representation, the definition of sites and sections is based on the following criteria:

1. Each site, whose dimensions may reach few kilometers, gathers assets with the same characteristics, or different assets collaborating to perform a unique task;
2. Sites gathering the same kind of assets, may be also differentiated with respect of their surrounding landscape (hills, plans, valleys), with respect of the presence of trees

Figure 1. Example of spatial discretization and the relevant graph representation



or houses, or, in general, with respect of the characteristics that influence the attractiveness of the site and the effectiveness of the protections;

3. Any section should not be longer than 3-5 kilometers.

An example of such a discretization is depicted in Figure 1(a) where the surrounding landscape is also indicated, when significant.

Moreover, they may be represented as in Figure 1(b), where the circles represent the elements constituting the railway infrastructure, whereas the arrows represent the following (qualitative) relationships among them:

- The continuous line between s_i and s_{i+1} indicates that when s_i becomes less attractive, then the risk of s_{i+1} increases;
- The dashed line between s_i and s_{i+1} indicates that when s_i becomes less vulnerable, then the risk of s_{i+1} increases.
- Elements not linked by continuous, dashed, of both the arrows may be thought of as completely independent from the point of view of attractiveness, vulnerability, or both of them, respectively.

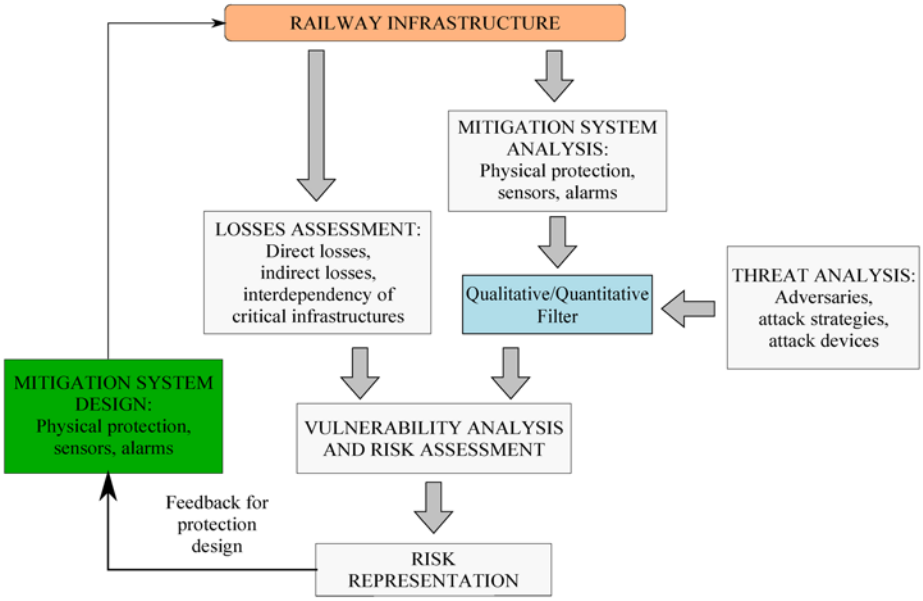
Note that such a discretization allows to represent a generic railway infrastructure as a set of physically distinguished elements that influence each other by means of a process described in the following.

To conclude it is worth saying that the representation introduced in Figure 1 (b), will constitute a basic model for understanding the risk of the whole railway infrastructure and will be detailed described from a quantitative point of view.

A General Architecture of a Risk Analysis Tool

In this section, the different modules that constitute a general architecture of a tool for the security risk analysis are introduced and discussed. To do so, consider the general scheme reported in Figure 2, where the different blocks have the meaning and perform the tasks described in the following:

Figure 2. General scheme for railway security risk analysis tools



- **The *Ground Transportation Infrastructure (TI)* Block:** Represents the ground transportation infrastructure, whose risk has to be assessed, and the relevant *Physical Protection System (PPS)* has to be designed;
- **The *Losses Assessment (LA)* Module:** Compute the damages generated by an attack, both in terms of direct losses (that is, the costs of all the assets destroyed or damaged by the attack), and of damages caused to other assets throughout the interdependence characterizing certain. The output of such a module consists of an evaluation of the whole damages in term of monetary costs, and of an estimation of the possible fatalities;
- **The *Mitigation System Analysis (MSA)* Module:** Analyzes, classifies, and evaluates the quality of the PPS already active in the considered infrastructure. The output of such a module consists of a list of the already operative PPS devices, as well as the evaluation of the relevant performances and effectiveness;
- **The *Threat Analysis (TA)* Module**: Identifies all the possible adversaries by taking into account the relevant characteristic such as typical targets, tactics, weapons, knowledge, and so on. The output of such a module consists of a list of the adversaries together with the most probable assets that could be attacked by each of them;
- **The *Vulnerability Analysis And Risk Assessment (VARA)* Module:** Represents the kernel of the whole architecture, and is devoted to the quantification of the risk for any asset of the considered railway infrastructure. Its inputs are the losses provided by the above LA module and the attack likelihood for any class of adversary, provided by the MSA and TA modules;

- **The *Risk Representation (RR)* Module:** Represents the part of the tool devoted to the analytical representation of the risk associated with each asset of the considered infrastructure. Its output corresponds to the so-called *risk profile*.

Looking at the above presented general architecture, it is worth noting that it requires further assumptions to be applied to railway infrastructures, which, as mentioned, consists of a set of different sections and sites with different characteristics. Then, assuming that all the elements in the set S are independent, the risk analysis of a whole infrastructure may be performed by applying the above scheme to each of them. Nevertheless, in doing so, the attractiveness of each site and section is evaluated by comparing all of them, thus individuating, among the elements in S , the elements with higher attack likelihood.

In addition, it is easy to note that the information provided by some of the above modules (TA and MSA) consist of qualitative descriptions of some characteristics, whereas the other modules require that such evaluations are expressed quantitatively (LA, VARA, RR). This is the reason why, in the scheme of Figure 2, proper *Qualitative-Quantitative Filters (QQF)* have been introduced in order to make the inputs and the outputs of each module compatible with each other, and the relevant conversion reliable. In doing so, it is worth pointing out that many approaches for such a conversion may be applied, depending on the kind of available data and on the target of the analysis. Examples of such filters may be conversion tables, for instance elaborated via brainstorming of security and railway experts, or more sophisticated models that elaborate the qualitative information by means of fuzzy logic.

To conclude, it is worth saying that when the aim of the risk analysis is to design an effective PPS, and in particular, when technical and/or budget constraints rise, a module performing an optimization of the available resources, hereafter indicated as *Mitigation System Design (MSD)* module, results to be of paramount importance.

The Threat Assessment Module

Quantifying the threats of malicious attacks is difficult task to tackle with, mainly due to the uncertainties that have to be taken into account. In this framework, the TA module performs the identification of the possible adversaries for any asset under analysis, also providing an evaluation of the relevant attractiveness. This module considers all the potential adversaries, their skills and equipment and identifies their profiles by means of a literature review or by means of the analysis of the events occurred in the world in recent years. To do so, the TA module assumes that each kind of adversary tries to carry out the attack having a detailed knowledge of the site. Note that such an assumption is not met in general but, in order to not underestimate the risk level, it is always a good approach to consider the worst case. Therefore, attackers are supposed to be suitably prepared and aware of the protection active in the systems. A list of possible adversaries is then generated, determining which assets are more attractive for each of them. Once identified these characteristics is hence possible to determine how easily each kind of opponent can overcome the various protections of the considered sites/sections throughout a QQF, as described in the following sections.

Quantitative/Qualitative Filter Module

The proposed architecture is designed for being used directly during the inspection of sites and sections, thus facilitating the risk assessment of large transportation railway networks.

Then, it is possible to note in the architecture of Figure 2 that the VARA module needs different kinds of data:

1. About potential damages and losses, that is, the cost of the repairing/rebuilding the damaged infrastructures, and cost depending on the system unavailability, such as the lost income due to the interrupted train circulation;
2. About the characteristics of the adversaries, such as tactics, objectives, and so on;
3. And regarding the protection systems, such as the kinds of detecting sensors, the kinds of physical protections, the number and the location of controlled accesses, and so on.

It is easy to note that while the first kind of inputs consists of numerical (economic) evaluations, the other are, on the contrary, typically expressed in a mixed qualitative/quantitative form.

In addition, security risk analysis is often based of parameters provided by experts that define tolerable risk thresholds, priority and weighting coefficients, and, in general, all the parameters that constitute the risk assessment process. Consequently, the need of converting and homogenizing such inputs from qualitative descriptions to quantitative evaluations rises, in order to allow the VARA module to compute numerical values of the risk.

To this aim, a Fuzzy Logic approach (Zadeh et al., 1968; Klir & Yuan, 1995) has been chosen due to its capability of *quantifying* the variables which are naturally qualitative such as the above mentioned asset descriptions, the expert opinions, and so on.

Therefore, in the following a fuzzy logic filter for estimating the attack likelihood of assets will be described, with the aim of providing an example of QQF.

Basic on Fuzzy Logic

In this section, the basic concepts for designing fuzzy logic filters for evaluating the attack likelihood and assessing the quality of the protections are discussed. Then, in order to make the contents of this section as clear as possible, it is necessary to briefly introduce the Fuzzy Logic (FL), and also analyzing possible alternative methodologies.

Then, as regards Fuzzy Logic, it was defined in 1965 by Zadeh as a mathematical tool for dealing with uncertainty, thus offering the important idea of computing with variables defined only by words. In effect, FL not only provides techniques to deal with vagueness and information granularity, but also a mechanisms for representing linguistic constructs such as *many*, *low*, *medium, often, few,* etc. Moreover, FL has been extended to handle the concept of *partial truth*, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, they can be managed by specific functions so that word expressions such as *age* may assume any values from *young* or its antonym *old*. However, the great utility of linguistic variables is that they can be modified via linguistic hedges applied to primary terms, by means of given functions. Then, once fuzzy relations are defined, it is possible to develop fuzzy relational databases that match data by using common characteristics found within the data set.

For what concerns the relation between FL and theory of probability, they are characterized by different approaches to the evaluation of uncertainty. While both the approaches can be used to represent subjective belief, FL uses the concept of fuzzy set membership functions measuring *how much* a variable is in a particular set, whereas probability uses the concept of subjective probability measuring *how probably* is a variable is in a particular set.

Then, before coming to the problem of designing QQF, it is worth summarizing that fuzzy logic guarantees the possibility of:

1. Representing linguistic constructs;
2. Translate such statements into more precise language, removing their semantic value;
3. Simulate the experts decision processes;

4. Obtain good results using a very limited source of information, if compared with the probability theory

As regards FL limitations in designing filters, it must be considered that:

1. It does not manage discrete variables;
2. It is sometimes hard to be adapted to a particular problem;
3. It does not always give definitive answers;
4. It is harder to program if compared with other approaches.

Design of a Fuzzy Logic Based QQF

The first step to be performed by the considered qualitative/quantitative filter is to collect the information about valuable assets, presence of people, and, in general, about the attractiveness of the asset that might be targets for some adversaries. In details, the characteristics to be analyzed are:

1. The locations with valuable materials, when thieves are considered;
2. The sites gathering many people, or those whose failures would have an impressive impact on the public opinion, when terrorists are considered;
3. The vital areas gathering those equipment whose failure, or damage, could make the railway services unavailable, when saboteurs are considered;
4. Isolated or unguarded places, when vandals are considered.

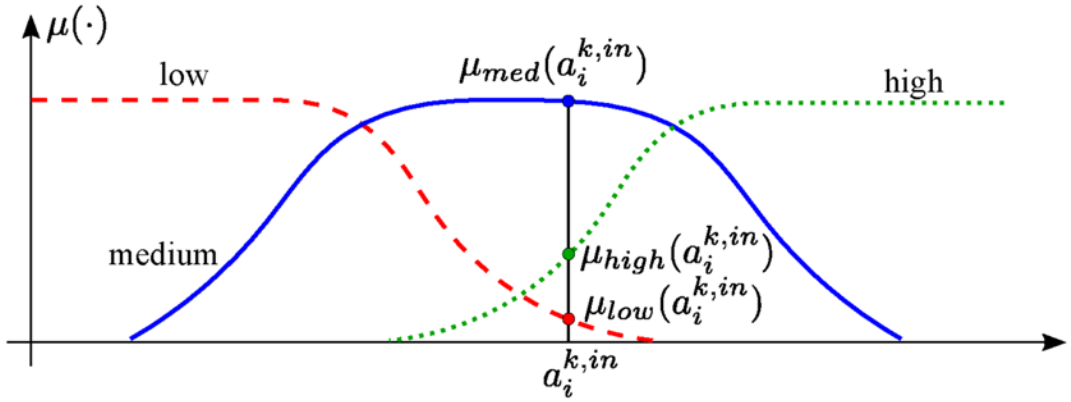
In doing so, the *risk assessor* is expected to give a qualitative score, hereafter indicated as $a_i^{k,in}$, in a scale between 0 and 10 and expressing the attractiveness of the $k-th$ asset with respect to the $i-th$ class of adversary. Therefore, the value $a_i^{k,in}$ is processed throughout the membership functions μ_{low}, μ_{med}, and μ_{high}, associated with the different qualitative values expressing a low, medium, and high attractiveness of assets, respectively, and whose shapes are reported in Figure 3. As usual in fuzzy logic, any asset belongs at a time to all the sets of lowly, medium, and highly attractive assets, although with different degree of membership.

Then, the fuzzy filter computes the value

$$a_i^{k,out} = a_i^{k,in} \cdot \Lambda$$
$$\Lambda = \mu_{low}\left(a_i^{k,in}\right) + \mu_{med}\left(a_i^{k,in}\right) + \mu_{high}\left(a_i^{k,in}\right) \quad (1)$$

which represents a weighted value of the input attractiveness $a_i^{k,in}$ that takes into account the

Figure 3. Membership function for "attractiveness of an asset"



experiences of the experts that have defined the membership functions.

Such a task may be obtained by collecting the scores about the attractiveness of some test assets provided by a different expert risk assessors, both in terms of numerical evaluation, that is the values $a_i^{k,in}$, and in terms of the qualitative association to one of the set gathering the lowly, medium, and highly attractive assets.

For what concerns the attack likelihood, it is computed by the QQF by means of a set of membership functions similar to those reported in Figure 3, which have to be calibrated for each class of adversary by means of ad-hoc questionnaires or field data.

Such functions relate the calibrated attractiveness $a_i^{k,out}$ of the $k-th$ asset with respect to the $i-th$ class of adversaries, thus providing an estimate the attack likelihood. Note that such a value has to be multiplied for the so-called a-priori attack probability, that is, the probability that provides, for all the assets, the generic attack likelihood that does not depend on the characteristics of the particular considered asset. Note that this parameter is of paramount importance and expresses the fact that in some countries the attacks of terrorists or of saboteurs are more frequent than in others, that in some cities vandals and thieves are more active than in the others, that some assets are more probable target than others, and so on. As regards the evaluation of such a-priori attack probability, it usually depends on the particular country homeland risk level, or on the particular city crime level, and it is usually by the Departments of Interior or by the Department of Defense.

The Mitigation System Analysis Module

The MSA module performs an analysis of the protection systems already in use in the considered site/section. By means of such an analysis, it is possible to perform the vulnerability analysis and estimate the effectiveness of protections.

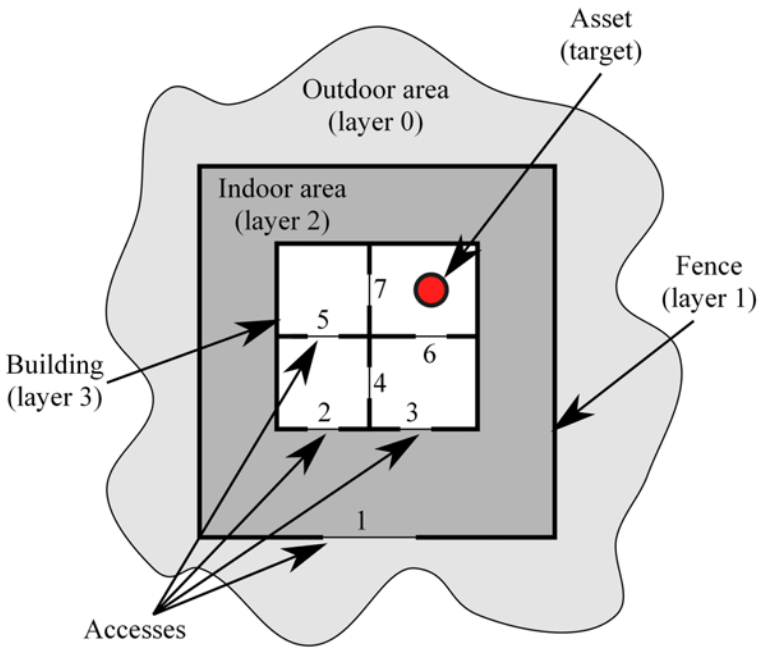
Then, consider the model of a generic site in Figure 4 where the interactions among the different components are explicitly represented (see Ustun et al., 2005 for an example of security simulation framework based on such a kind of representation). In such a model, it is easy to identify three distinct areas around the site, as well as the relevant accesses and PPS. In particular, in such a scheme, the following areas may be identified:

1. **Border Areas:** The area covered with fences between the site under analysis and the external environment. The information characterizing such an area are:
   a. The percentage of the perimeter of the structure defended using physical protections and those protected using technological devices (video-cameras, active fences, and so on);
   b. The accessibility of the site/section.

In this framework, the kind and the efficacy of protections and of the access, defined as their robustness against an attack, can be evaluated directly by risk assessors, and allow to assess the degree of protection for border areas, the capability of detecting an attack, and the kind of equipment necessary to carry it out;

2. **Outdoor Areas:** The areas not enclosed within a building or fence with protected access. In such areas three different kinds of information are considered:
   a. The operating space available for vehicles which is closely related to the type of vehicle that could be used to commit the attack;
   b. The percentage of square covered by sensors;
   c. The presence of possible attractive elements for adversaries;

Figure 4. Example of site with the relevant areas and protections



3. **Indoor Areas:** The areas located inside a building or a fence. With respect to these areas, the number and the kind of accesses are considered, also taking into account the degree of sensing coverage. The analysis is completed by the individuation of the exact position of asset to be protected.

All this information collected by such a module may be modified in order to evaluate possible increase or decrease of protection level, in order to find the most effective one.

Then, as regards the numerical evaluation of the PPS effectiveness, it mainly consists of the estimation of the *probability of identifying an attack as soon as possible*, thus allowing to react and stop the adversaries.

Then, while in some cases, the PPS effectiveness is provided as a design characteristic of the protection itself, in other cases, it has to be estimated by the risk assessors.

Examples of effectiveness provided as design characteristics are the probability of individuating intruders in open areas by means of the elaboration of images provided by digital Closed Circuit Tele-Vision (CCTV), or the cross of electrostatic virtual fences by means of capacitance based boundary sensors, and so on (see Garcia, 2001, for a detailed description of several sensor technologies and of their effective use). On the other hand, examples of parameters to be estimated may be the effectiveness of different kind of walls and fences, that may discourage or not a potential adversary. In this last case, in analogy with the attractiveness parameter above described, these qualitative evaluations can be converted into *detection likelihoods* by means of FL filters. In particular, each value $q_i^{j,k,in}$ assigned by a risk assessor to the $j-th$ protection of the $k-th$ asset with respect to the $i-th$ class of adversaries, is converted into a numerical value $q_i^{j,k,out}$ throughout suitably shaped membership functions, as described in section 2.4.2.

To conclude it is worth underlining that, in railway infrastructures, some sections are highly accessible and scarcely protected, whereas other are fenced and protected by sensors and alarms. Examples of such the first kind of sections and sites are railway stretches or power distribution, whereas examples of the second ones are command and control centers, and any kind of signaling interlocking, communications and information critical system.

The Vulnerability Analysis and Risk Assessment Module

As previously said, the VARA module combines the information provided by the LA module and by the TA and MSA modules (the last two throughout the QQF) and computes the risk value for each asset of the considered infrastructure.

Then, in this section, the models for computing the risk of an isolated and coupled sites and sections are presented. Since such an analysis is based on elementary Graph Theory, the interested reader may refer for further details to (Bondy & Murty, 1980).

Isolated Site/Section

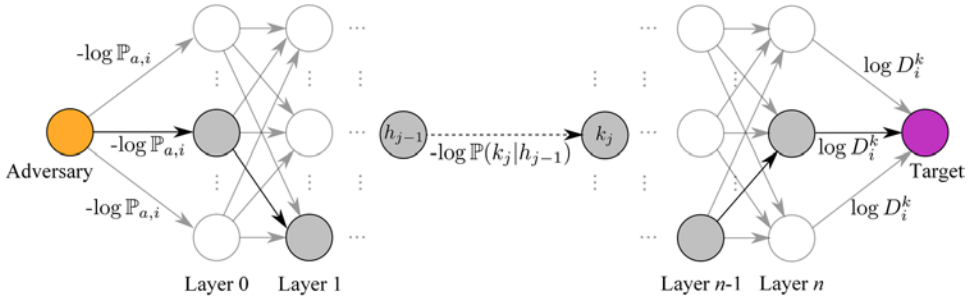
Consider again the scheme reported in Figure 4: any path from the outdoor area towards the target must cross the areas described in the section 2.5, the relevant perimeters (if the areas are protected by fences or walls), and overcome a set of protections securing them.

Then, such possible adversary paths for reaching a generic asset x can be thought of as paths throughout different protection layers j, $j=1,\ldots,n-1$, each characterized by a set of n_j possible accesses h_j, $h=1,\ldots,n_j$, or k_j, $k=1,\ldots,n_j$. In this representation, the layer 0 is represented by the adversary, whereas the layer n is represented by the target asset x.

Then, consider the graph reported in Figure 5, where the arc from the node h_{j-1}, $j=1,\ldots,n-1$, to the node k_j, $j=1,\ldots,n-1$, has weight equal to the cost to overcome the k_j-th access of the layer j coming from the $h_{j-1}-th$ one. In other words, the cost of any arc $\left(h_{j-1},k_j\right)$, $h=1,\ldots,n_{j-1}$ $k=1,\ldots,n_j$, $j=1,\ldots,n-1$, represents a measure of the easiness to reach k_j, $k=1,\ldots,n_j$, coming from h_{j-1}, $h=1,\ldots,n_j$, that is the relevant vulnerability. Then, from the point of view of an adversary, arcs with high costs are preferable, since represent much vulnerable paths.

Note that the cost of the arcs $\left(h_{j-1},k_j\right)$ and $\left(l_{j-1},k_j\right)$, $l\neq h$, are usually different. As an example of such a phenomenon, suppose that the node k_j represents the door of a building that has to be reached by an adversary as soon as possible. Assume also that whole outdoor area protected by means of a CCTV. Evidently, entering the protected area throughout the nearest fence to the

Figure 5. Graph of the admissible adversary path



door (i.e., the node h_{j-1}) facilitates the adversary with respect to any other more distant access (i.e., the nodes l_{j-1}, $l \neq h$), since in the first case the probability to be detected is minimized.

Other examples of such measures may be the delay due to the need of overcoming the protections of an access, of crossing an outdoor area, of opening a locked door, or the probability being individuated by sensors when trying to overcome a protection.

In addition, the cost of the same arc depends on the equipment and on the skills of the considered adversary, previously identified by the above described TA module whose results have to be taken into account. Then, this is the reason why the dependence on the $i-th$ class of adversaries must be considered in the model of Figure 5.

To complete the graph, with respect to any $i-th$ adversary, the costs from the node *Adversary* to all the nodes of layer 1 are equal and represent the attack probability, whereas the costs from all the nodes of the $(n-1)-th$ layer to the node *Target* are equal and represent the damage provoked by a successful attack.

For what concerns the evaluation of the adversary path, that is the most vulnerable one, assume that in the graph of Figure 5, in place of the probabilities above defined, the following co-logarithms are considered:

1. The co-logarithm $-\log \mathbb{P}\left(h_{j-1} \mid k_j\right)$ instead of the probability $\mathbb{P}\left(h_{j-1} \mid k_j\right)$;
2. The co-logarithm $-\log \mathbb{P}_{a,i}^{x}$ instead of the probability $\mathbb{P}_{a,i}^{x}$ that the $i-th$ adversary attacks the asset x;
3. The logarithm $\log D_i^x$ instead of the loss D_i^x, which is greater than 0 if the loss D_i^x is suitably expressed in a scale whose minimum is 1.

With these choices all the costs are positive, and the more probability $\mathbb{P}\left(h_{j-1} \mid k_j\right)$ of an arc is near to 1, the more the relevant co-logarithm is near to zero, and the more the probability $\mathbb{P}\left(h_{j-1} \mid k_j\right)$ is near to 0, the more the relevant co-logarithm is greater than 0.

Then, since any adversary tries to maximize the vulnerability of the path, it chooses the *longest path* $P_{\max}$, which is the one with maximum vulnerability, i.e., the one with minimum detection probability, which is characterized by the cost

$$\begin{aligned} C_{\max} &= -\log \mathbb{P}_s^k \\ &= \sum_{\left(h_{j-1},k_j\right)\in P_{\max}} -\log \mathbb{P}\left(h_{j-1},k_j\right) \end{aligned} \tag{2}$$

being $\mathbb{P}_s^x$ the probability if completing an attack successfully, when tried. Then, the detection probability results to be

$$\mathbb{P}_s^k = \mathbb{P}\left\{\text{success}|\text{attack tried}\right\} = e^{-1/C_{\max}} \tag{3}$$

Finally, it is possible to state the probability of a successful attack by means of the two equivalent formulas

$$\begin{aligned} \mathbb{P}_i^x &= \mathbb{P}\left\{\text{success}|\text{attack tried}\right\} \\ &\quad \cdot \mathbb{P}\left\{\text{attack of adversary } i\right\} \\ &= \mathbb{P}_{a,i}^x \cdot \mathbb{P}_s^x \\ \mathbb{P}_i^x &= e^{-\log \mathbb{P}_s^x - \log \mathbb{P}_{a,i}^x} \end{aligned} \tag{4}$$

whereas the risk R_i^x of the $x-th$ asset with respect to the $i-th$ class of adversaries, is simply given by the two equivalent formulas

$$\begin{aligned} R_i^x &= \mathbb{P}_{a,i}^x \cdot \mathbb{P}_s^x \cdot D_i^x \\ R_i^x &= e^{-\log \mathbb{P}_{a,i}^x - \log \mathbb{P}_s^x + \log D_i^x} \end{aligned} \tag{5}$$

Finally, as regards the risk of the $y-th$ generic section or site gathering several different

assets, it corresponds to the risk of the asset with the maximum risk, that is,

$$R_i^y = \max_{x \in y} R_i^x \tag{6}$$

To conclude, it is worth underlining that with this representation it is easy to understand that introducing new devices that reduce the vulnerability of one or more accesses, or areas, may not necessarily correspond to a reduction of the risk, unless the cost of the longest path (expressed in terms of co-logarithmic costs) is reduced. In other words, if the cost of the so-called second best, third best, and so on, paths are lesser but near to $C_{\max}$, then introducing protections does not provide a considerable risk reduction.

Coupled Sites and Sections

Consider two or more different sites/sections that influence each other as explained in section 2.1.

The risk of any of these sites does not depend only on the characteristics of the site/section itself, but also on the characteristics of the others. Such a mutual dependence is mainly due to the interactions between the attractiveness and the vulnerability of different assets, and may be exploited by considering that the attractiveness of any asset also depends on its vulnerability, that is, in other words, $\mathbb{P}_{a,i}^x$ depends on $\mathbb{P}_s^x$. Then, reducing the term $\mathbb{P}_s^x$ may cause a change of interests of some adversaries, which would attack another more vulnerable asset, that is, in other words, it may increase the probabilities $\mathbf{P}_{a,i}^z$, $\forall z \neq x$.

Again, modeling all the possible adversary paths throughout a single graph allows exploiting such a phenomenon. Then, consider the graph in Figure 6 where the areas and the PPS of two sections/sites, characterized by similar potential loss, are considered. In this case, being the damages equal for both the targets, the longest path between the two from the node adversary to the nodes target 1 and target 2 indicates the most probable target for the considered adversary, that is the one that is more easily reachable without being individuated.

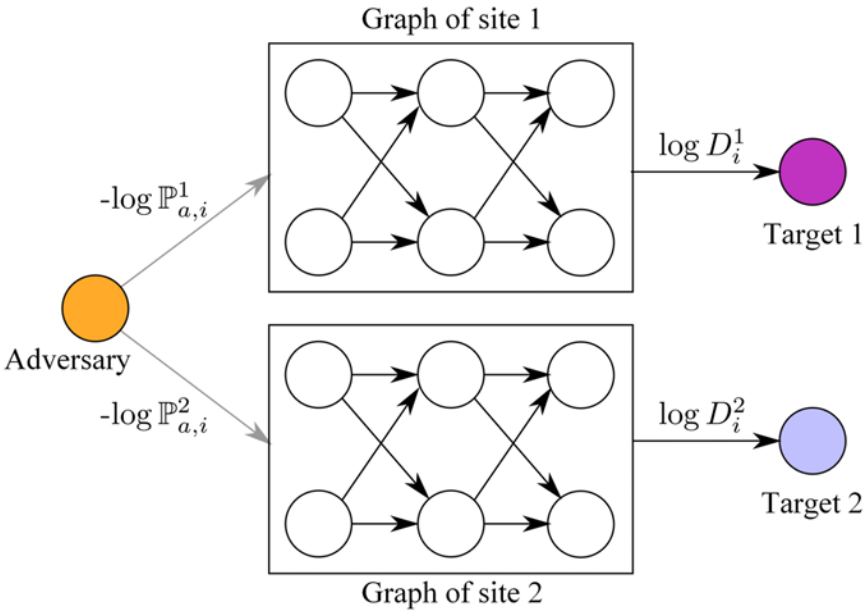
As previously mentioned, suppose that the target 1 is the most probable objective of the adversary. Then, assume that all the protections of such a site/section are improved, thus significantly reduce the costs (expressed as co-logarithms) of all the paths from adversary towards target 1. If such an improvement makes one of the possible paths from adversary to target 2 the longest one, than the attacker changes its objective.

Such a phenomenon figures out that any PPS improvement need to be carefully chosen so as to avoid great investments that simply moves the attention of potential adversaries, without really reducing risk. In other words, the investments should be carefully evaluated to globally reduce the risk level of the whole considered transportation infrastructure. To do so, in the following section 2.8, an optimization problem will be stated in order to reduce the risk of all the sites and sections, in opposition to the problem of considering each of them separately.

Risk Representation Module

In this section, some considerations about the risk representation for railway infrastructures are discussed. To this aim, consider again the general architecture depicted in Figure 2. When the risk of each site/section of the railway line under analysis is assessed, it is possible to draw a *risk profile diagram* by simply reporting on one axis the distance of the site/section from the initial one, and on the other axis the relevant risks for any class of adversaries. Evidently, such a representation (whose example is reported in Figure 1 (c) has the advantage to show, at a glimpse, how

Figure 6. Graph of adversary path of two assets with comparable potential damage



risk varies along the railway line, thus allowing to easily identify all the sites/sections characterized by intolerably high-risk values. To do so, it is sufficient to identify a threshold line that indicates the maximum admissible risk value for any site/section. Therefore, if some sites/sections have a risk value, for a certain kind of threat, higher that the threshold, then these sites need investments to increase the relevant PPS effectiveness.

Once such sites/sections are identified, the Mitigation System Design (MSD) module in Figure 2, proposes PPS the improvements for the considered infrastructure throughout a *feedback* process. Then, the previously described VARA module assesses the new infrastructure configuration, and provides a new risk profile: if it results to be acceptable, then the PPS design process is finished, otherwise the MSD module has to provide further, or alternative, improvements.

Optimal Mitigation System Design Module

In this section, an investments optimization problem based on the results of the above described modules is introduced. In doing so, it is worth recalling that securing a railway infrastructure has to be faced as a global problem that, in principle, must take into account at a time all the site/section constituting the railway network.

To this aim, let c_h be a vector whose generic element is set to 1 is the relevant intervention is activated, and to 0 otherwise. With this definition, the dimension K of the vector correspond to the number of possible interventions for the considered railway infrastructure, whereas the vector $c_0 = \begin{bmatrix} 0 & \dots & 0 \end{bmatrix}^{\mathrm{T}}$ represents the initial configuration without any intervention, and $c_N = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}^{\mathrm{T}}$ is associated with the configurations in which all the interventions are activated. It is easy to observe that there are $N = 2^K$ possible values of vector c_h hereafter considered to be gathered in the set $C = \{c_h \mid h = 1, \dots, N\}$.

With these assumptions, the optimization problem may be stated as

$$\begin{aligned} &\min_{c_h \in C} J \\ &\text{s.t.} \\ &\mathrm{c}_h W^T \leq c_{\max} \qquad \forall \mathrm{c}_h \in C \end{aligned} \tag{7}$$

where $c_{\max}$ is the maximum budget available for interventions, and $w_N = \begin{bmatrix} w_1 & \dots & w_N \end{bmatrix}^{\mathrm{T}}$ is the vector gathering the costs of all the interventions.

As regards the cost function, it expresses the sum of the risks for all the sites/sections and for all the possible adversaries, that is

$$J = \sum_{\forall i \in I} \sum_{\forall x \in S} \alpha_i R_i^x \tag{8}$$

where I is the set of all the classes of adversaries, and α_i, $\forall i \in I$, is a suitable weighting term introduced to differentiate importance of the different classes of adversaries.

Note that the correlation among the risks of all the assets is hidden in the risk values R_i^x which have been assessed by means of the model above presented. As regards the problem solution, when the number of variables K is small enough (less than $5 \div 7$ possible interventions), a brute force searching approach is suitable by enumerating the possible $N \leq 32 \div 128$ configurations. On the contrary, for greater values of K, genetic algorithms (Srinivas & Patnaik, 1994; Whitley, 1994; Banković Z., 2007) appear to be a suitable methodology to find the best configuration. In fact, it is easy to note that the problem in Equation (7) and Equation (8) has a formulation that suits to this class of algorithms, being the vectors $c_h \in C$ suitable genetic codifications of the solutions.

CASE STUDY

In this section, a case study based on a real world case is presented. To this aim, after a brief description about the considered assets, the results of the risk assessment obtained by means a software designed on the basis of the proposed architecture are described. Then, with the aim of pointing out the importance of the optimization problem defined in Equation (7) and Equation (8), the effects of the introduction different PPS in the considered railway line are discussed.

Note that the presented case study is based on a risk assessment activity performed by the authors on the Italian high-speed railways, and then, for security reasons, some information about the considered assets is necessarily dropped.

The Considered Railway Line and Threats

The considered case of study consists of a short railway stretch characterized by a sequence of 10 sites/sections chosen among the following ones:

- Tunnels, as the one depicted in Figure 7;
- Bridges, as the one depicted in Figure 8;
- Open-Air Tracks, as the one depicted in Figure 9;
- Power Supply Plants, as the one depicted in Figure 10.

Then, while the main characteristics of such sites/sections are summarized in Table 2 together with the surrounding characteristics that influence the site/sections accessibility, it is worth saying that no particular physical protection systems for security risk mitigation are considered in the initial risk assessment. As regards the considered adversaries, the attention is indeed focused on terrorists (Te), thieves (Th), and vandals (V).

Figure 7. Example of a tunnel



Figure 8. Example of a bridge



Figure 9. Example of open air track



Risk Assessment

For what concerns the risk assessment, the information provided for the considered railway stretch by the risk assessors are reported in Table 3, where also the attack likelihood, computed by the QQF, is reported for the different classes of adversaries.

As regards the highest attack likelihood values, it is worth saying that:

- Bridges are extremely vulnerable the easiness for vandals to make graffiti on pillars;
- Thieves may act quite undisturbed in power supply plants and especially along the railway line. This is the characteristic that makes copper steal almost frequent along railways;
- Tunnels and bridges are quite attractive for terrorist and saboteurs due to high impact of the relevant damages, both in term of economic losses and fatalities.

Finally, taking into account the mean, direct and indirect, damages caused by the considered classes of adversaries to the considered assets, it is possible to compute the risk and build the risk profile depicted of Figure 11, where the risk values are expressed, for all the considered railway stretch and threats, in monetary costs.

Risk Mitigation Design

As mentioned, of the MSD module evaluate the possibility to optimize the interventions aiming to globally reduce the risk of all the considered railway stretches. Then, to the end of showing the tasks performed by such a module, in this section the effects of the introduction of a CCTV system for monitoring the power plant, the bridge pillars and the tunnel access are described. In doing so, it has not been possible to design an economically sustainable CCTV plant able to control the entire open air track site, which remains, indeed, unchanged.

Figure 10. Example of Electric power supply plant.



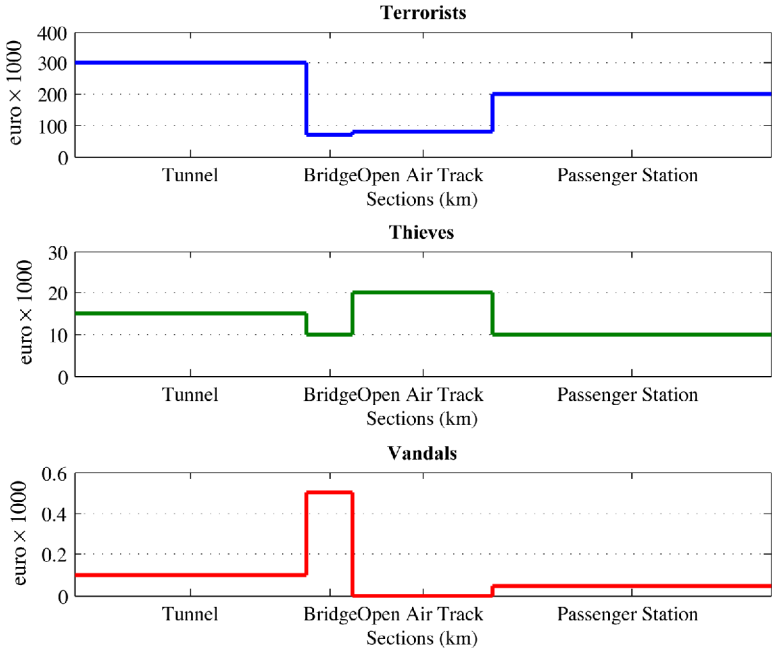
Table 2. Sites making up the considered railway stretch

Section		Length of section [km]	Surrounding landscape and outdoor area
1	Open air section	3.5	Country
2	Open air section	3.5	Hills
3	Tunnel	0.5	Hills
4	Open air section	2.7	Hills
5	Power supply plant	0.15	Country
6	Open air section	3.2	Tree
7	Tunnel	0.45	Hills
8	Bridge	0.2	Hills
9	Open air section	1.5	Hills
10	Open air section	3.5	Tree

Table 3. Characteristics of the sites and attack likelihood

Section	Attractiveness			Attack Likelihood (times/year)		
	Te	Th	V	Te	Th	V
Tunnel	6	6	2	10E-8	10E-6	10E-4
Bridge	2	2	7	10E-9	10E-8	8
Open air section	2	7	2	10E-5	10E-1	10E-5
Power supply plant	9	1	1	10E-3	3	10E-9

Figure 11. Level of risk of each section for each type of adversary



In addition, the introduced PPS are assumed to be visible by anyone, since such a characteristic significantly influences the attractiveness of a site (Klir & Yuan, 1995). In effect, the visibility of the CCTV devices discourages adversaries to attack thus reducing the terms $\mathbb{P}^{x}_{a,i}$ for all of the protected sites.

Then, a comparison between the risk profile of the considered railway before and after the introduction of the CCTV is reported in Figure 12 where it is easy to observe that:

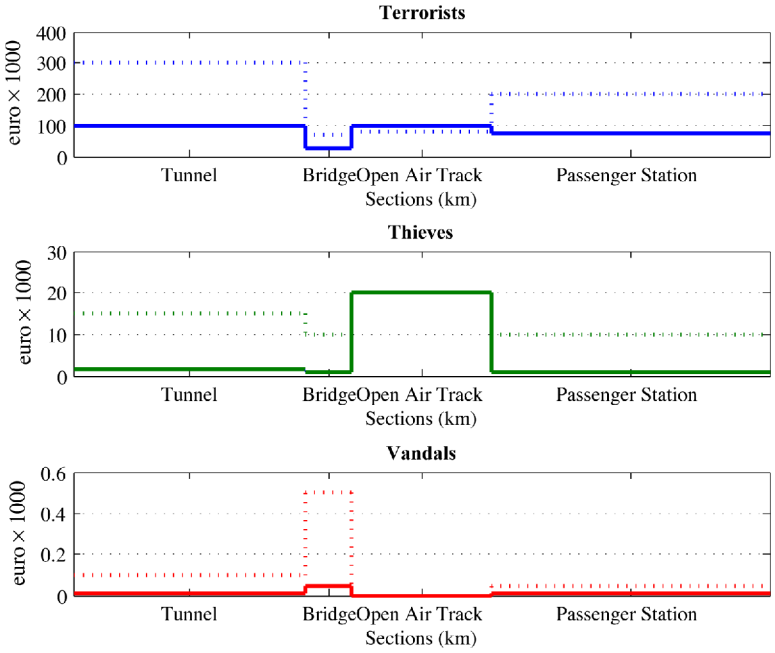
- For what concerns the attacks of terrorists, the introduction of the considered PPS reduces, on one hand, the risk in the protected sites, but increases, on the other hand, the risk of the open air track sections;
- The risk computed with respect of thieves and vandals, is reduced in the protected sites, but remains unchanged, although the last one is negligible, for the open-air track sections.

Such an effect points out the previously mentioned dynamics in PPS design: securing a single asset does not necessarily make the whole railway system more secure, because the mitigation actions might only move the interest of adversaries towards less protected assets.

In terms of the cost function defined in Eq. (9), where for the sake of simplicity the weights α_i has been all set to 1, the entire risk decreases from 705.65k€ to 323.57k€.

Then, the capability of the proposed architecture of correlating the effects of changes on individual sites/sections and to reassess the level of security of the whole infrastructure throughout a feedback process is hence showed. Such a capability, combined with the possibility of trying different kinds of securing interventions, is suitable for the best choice of investment and constitutes the way for evaluating the cost function in Eq. (9) for each candidate solution independently from the chosen optimization algorithm.

Figure 12. Comparison of the risk profiles before (dashed lines) and after (continuous lines) the introduction of a CCTV



CONCLUSION

In this chapter, a general architecture for designing risk analysis tools for railway system security has been introduced. In particular, in addition to the description of the modules making the architecture up, a particular attention has been focused on the two major problems that make security risk analysis of distributed infrastructures a difficult task to tackle with:

1. The qualitative nature of part of the data involved in the risk analysis process;
2. The interaction of different assets, which may be associated in terms of their locations, but also in terms of their comparable attractiveness, consequent potential losses, and so on.

As regards the first problem, the different kinds of data exchanged by the different architecture modules as input and/or output have been discussed in detail. Then, to cope with the problem of making such data comparable and reliable, a fuzzy logic based approach has been proposed as the kernel of a qualitative/quantitative filter to be used as an interface between the different modules.

For what concerns the second problem, a suitable analysis approach based on a graph representation of the targets and of the PPS has been introduced and discussed. By means of this representation, it has been possible to figure out the interactions among different site/section, and then stressing out the need of an optimization module.

To conclude, a case study has been presented with the aim of proving the effectiveness of the proposed architecture.

REFERENCES

Akgun, I., Kandakoglu, A., & Ozok, A. F. (2010.), Fuzzy integrated vulnerability assessment model for critical facilities in combating the terrorism, in *Expert Systems with Applications, 37*.

Amin, M. (2002), Toward secure and resilient interdependent infrastructures, in *Journal of Infrastructure Systems, 8*, 67–75.

Apostolakis, G. E., & Lemon, D. M. (2005), A Screening Methodology for the Identification and Ranking of Infrastructure Vulnerabilities Due to Terrorism, in Risk Analysis, in *Risk Analysis, 25*, (2), 361-376.

ASIS International. (2003). *General Security Risk Assessment Guideline*. Alexandria, VA: ASIS International.

Aven, T. (2007), A uni□ed framework for risk and vulnerability analysis covering both safety and security, in *Reliability Engineering and System Safety, 92*, Elsevier.

Bajpai, S., Sachdeva, A., & Gupta, J. P. (2009). Security risk assessment: Applying the concepts of fuzzy logic. In *Journal of Hazardous Materials*. Elsevier. doi:10.1016/j.jhazmat.2009.08.078

Banković, Z., Stepanović, B., Bojanić, S., & Nieto-Taladriz, O. (2007). Improving network security using genetic algorithm approach, in *Computers &. Electrical Engineering*, *33*, 438–451.

Banks, J. J., & Carson, S. Nelson, B. L. &,Nicol, D. M. (2000), *Discrete-Event System Simulation. (3rd Ed)*, Upper Saddle River, NJ: Prentice Hall.

Bondy, J. A., & Murty, U. S. R. (1980). *Graph Theory with Applications*. New York, NY: North-Holland.

Cassandras, C., & Lafortune, S. (2008). *Introduction to Discrete Event Systems* (2nd ed.). Springer Science. doi:10.1007/978-0-387-68612-7

Commission, E. U. (2005). *Council Directive 2005/65/EC of October 26th 2005 on enhancing port security*. Official Journal of the European Union.

Commission, E. U. (2008). *Council Directive 2008/114/EC of December 8th 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection*. Official Journal of the European Union.

Commission, E. U. (2010). *Council Regulation 2010/186 of March 4th 2010 laying down detailed measures for the implementation of the common basic standards on aviation security*. Official Journal of the European Union.

Di Febbraro, A., Papa, F., & Sacco, N. (2010), A Tool for Risk Analysis and Protection Design of Railway Infrastructures, in *Proceedings of 89th TRB Annual Meeting*. TRB.

Di Febbraro, A., & Sacco, N. (2010*), A Petri-Net based approach for the interdependence analysis of Critical Infrastructures in transportation networks*, in *Proceedings of 12th World Conference on Transportation Research*. COTA.

Eames, D. P., & Moffett, J. (1999), The Integration of Safety and Security Requirements, in *Proceedings of the 18th International Conference on Computer Safety, Reliability and Security,* (, pp. 468-480) Springer Verlag

Farrow, S. (2004), Using Risk Assessment, Benefit-Cost Analysis, and Real Options to Implement a Precautionary Principle, in *Risk Analysis, 24*.

Fink, C. N. Y. (2003), Antiterrorism Security and Surface Transportation Systems - Review of Case Studies and Current Tactics, in *Journal of the Transportation Research Board, 1822*, 9–17.

Garcia, M. L. (2001). *The design and evaluation of physical protection systems*. Elsevier.

Garrick, B. J., Hallb, J. E., Kilgerc, M., McDonaldd, J. C., O'Toolee, T., Probstf, P. S., et al. (2004*), Confronting the risks of terrorism: making the right decisions,* in *Reliability Engineering & System Safety, 86*, 129-176.

Hartong, M., Goel, R., & Wijesekera, D. (2008), Security and the US rail infrastructure, in *International Journal of Critical Infrastructure Protection, 1*, 15-28.

Hong, X., & Dugan, J. B. (2004), Combining dynamic fault trees and event trees for probabilistic risk assessment, in *Reliability and Maintainability, 2004 Annual Symposium - RAMS*, (pp. 214- 219.)

Hood, J. N., Olivas, T., Slocter, C. B., Howard, B., & Albright, D. P. (2003), Vulnerability Assessment Through Integrated Transportation Analysis, in *Journal of the Transportation Research Board, 1822*, 18–23.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer.

Kaplan, S. (1992), Expert information versus expert opinions: another approach to the problem of eliciting/combining/using expert judgment in PRA, in Journal *of Reliability Engineering & System Safety, Elsevier, 35*, 61-72

Kaplan, S., & Garrick, G. J. (1981). On the quantitative definition of risk. In *Risk Analysis* (pp. 11–27). Wiley.

Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic*. Prentice Hall PTR.

Lambert, J. H., & Farrington, M. W. (2007), Cost-benefit functions for the allocation of security sensors for air contaminants, in *Journal of Reliability Engineering & System Safety, 92*, 930-946.

Lee, W. S., Grosh, D. L., Tillman, F. A., & Lie, C. H. (1985), Fault Tree Analysis, Methods, and Applicationn: A Review, in *IEEE Transactions on Reliability, 34*(3),194-203.

Murata, T. (1989), Petri Nets: Properties, Analysis and Applications, in *Proceedings IEEE, 77*. 541-580.

Murray-Tuite, P. M. (2007), Transportation Network Risk Profile for an Origin-Destination Pair: Security Measures, Terrorism, and Target and Attack Method Substitution, in *Proceedings of 87th TRB Annual Meeting.*TRB

Peterman, D. R. (2006). Overview of Issues. In *CRS Report for Congress*. Passenger Rail Security.

Raj, P. K., & Pritchard, E. W. (2000) Hazardous Materials Transportation on US Railroads, in *Transportation Research Record, 1707*.

Ravi, V., Redd, P. J., & Zimmermann, H. (2000), Fuzzy global optimisation of complex system reliability, in *IEEE Transactions on Fuzzy Systems, 8*(3), 241–248.

Rowshan, S., Sauntry, W. C., Wood, T. M., Churchill, B., & Levine, S. R. (2005). *Transportation Research Record, 1938*. Reducing Security Risk for Transportation Management Center.

Srinivas, M., & Patnaik, L. M. (1994), Genetic algorithms: a survey, in *Computer, 27*, 17-26.

US Congress, (2007), *S. 184, The Surface Transportation and the Railway Security Act of 2007*.

Ustun, V., Yapicioglu, H., Gupta, S., Ramesh, A., & Smith, J. S. (2005), A Conceptual Architecture for Static Features in Physical Security Simulation, in *Proceedings of the 2005 Winter Simulation Conference.*

Zadeh, L. A. (1968), Fuzzy algorithms, in *Information and Control*.

Zeng, D., Chawathe, S. S., Huang, H., & Wang, F. (2007), Protecting Transportation Infrastructure, in *IEEE Intelligent Systems, 22*, 8-11.

Section 8

Experiences and Case-Studies

Chapter 18
ETCS Developing and Operation: Italian Experience

Raffaele Malangone
RFI, Italy

Fabio Senesi
ANSF, Italy

ABSTRACT

ETCS/ERTMS actually is the present for Italian Railways but it will also be the next future for the signalling system in many countries and the best technological choice for ATC (Automatic Train Control) systems. Italian Railways, first in the world, have carried out ETCS Level2 merging the technologies and the regulations respecting the highest safety level. RFI, following the CENELEC 50-126 Lyfe-Cycle, has developed a process for planning, managing, monitoring and controlling of ETCS achievements. In particular the Disposal 29 and 32 in the year 2002 have been issued for the assessment and homologation process of Generic and Specific Applications and other following procedures have permitted the final configuration of the project until its putting in service. The goal of this course is the Preliminary Acceptance of a Generic Application and later, after a successful testing period on field, its Homologation. RFI has followed the developing process starting from the idea to define the specifications, evaluating the hazards and their probabilities, finding mitigations to improve the safety, validating the products of the suppliers, testing the subsystems and the entire system, until the final activation of the whole systems (compliant with the Technical Specifications for interoperability, UNISIG v.2.2.2).Much attention has been paid to the testing of functional scenarios (using also formal languages) and the real tests on the track have been reduced with the support of an ERTMS Laboratory in Rome (unique in the world for its characteristics) where on-board and track-side subsystem permit to reproduce easily and quickly most of the real situations.

DOI: 10.4018/978-1-4666-1643-1.ch018

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

This testing process in ETCS laboratories has been useful not only before the putting the ETCS in service but also for the reconfiguration of the actual ETCS lines as it would be hard to do so many test scenario during a commercial service. These activities have been replicated several times, for example, to reach the actual ETCS version compliant to the UNISIG 2.3.0d. The success of the formal language analysis of Test-Specifications has also encouraged the RFI ETCS group to develop a state-charts model of the functional specification. This work is actually in progress but a first result, on the logical behavior of the system at the transition with a historical signalling system, has been done and validated.

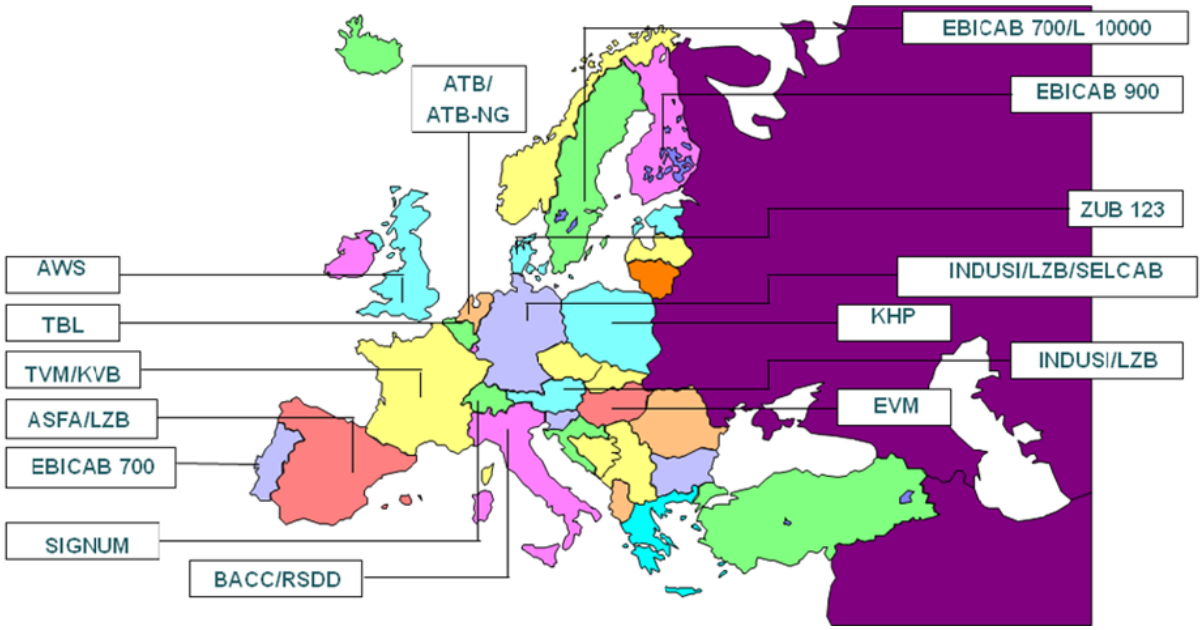
INTRODUCTION

Rail transport is a strategic historical sector in the worldwide policy for a sustainable mobility (Obama, 2008) (European Commission, 2010) (European Commission, 2006). In Europe, for instance, one of the main effort for this scope has been the developing of the Trans-European projects for railway interoperability with the aim to avoid the saturation of certain major arteries, and, thus, the related pollution, and to support transport modality with a lower environmental impact, removing, for instance, the bottlenecks in the railway network and supporting the develop of an over-national railway network.

In fact, although most of the rolling stock is technically capable to travel on a wide part of the European rail network, today the same cannot be said of locomotives, which suffer numerous constraints concerning the different electrification and signalling systems (Figure 1 Signalling systems in European countries) at the borders of the national systems (European Commission, 2010) .

To overcome this problem, in 2004 the European Commission has issued the Directive 2004/50/EC that defines the conditions to achieve interoperability in order to guarantee the safe and uninterrupted movement of trains crossing two systems without any performance reduction The directive concerns both the high speed lines and

Figure 1. Signalling systems in European countries



the conventional ones, merging the Directives 96/48/EC and 2001/16/EC respectively referred to the Trans-European high speed and the Trans-European conventional rail, respectively. This process has been confirmed by the issue of Directive 2006/860/EC, amended by Directive 2007/153/EC, referred to the European Train Control System (ETCS), that will be progressively installed in all European interoperable infrastructures and rolling stock, substituting the current "national" systems. In particular, in the aforementioned Directive, ETCS has been defined in a set of specifications that are referenced in the Technical Specifications for Interoperability for the Control-command and Signalling sub-system.

In this scenario the RFI – Rete Ferroviaria Italiana – the national company that manage the railway infrastructure including the signalling and interlooking subsystems – in 2005 has carried out the first world ERTMS/ETCS Level 2 on a line (the Roma-Napoli) without a fall-back signalling system and in the following years it has completed about 900 kilometres using this technology (Senesi, 2008) (Senesi, 2007b) (Senesi, 2010).

Italy has been the first country where ETCS Level 2 has been carried out for commercial.

This paragraph wants only to be a summary of the arguments treated otherwise too many pages should be necessary, but all the details can be found in the bibliography references.

TECHNOLOGY AND INNOVATION FOR IMPROVING RAILWAYS

Equipments and Systems for Railways and Metros have always profited by technological progress, that has permitted a continuous improving of the safety, of the functionalities for traffic management and increasing capability.

Most of Railways Systems have a quite long life-cycle (generally 20 years), due to the carrying out times and to the returns from the fields that a costly investment must ensure; this is incompatible with the frantic technological development in the last fifteen years that continuously provides new solutions.

The slow pace of technological upgrade therefore determines the following problems:

1. Low functional, plant and technological homogeneity;
2. Huge technological leaps in cases of abandonment and replacement of plant;
3. Difficulty of interfacing between components, equipment and subsystems that, even if coexisting, are designed, manufactured and introduced in very far periods.

For all this, there has been in recent years the effort of Italian Railways to modernize its network both with high-speed line and with new computerize systems for safety on historical lines.

If we think that a few decades ago there were no good previsions for the future of rail system, we can already be met.

It was not appreciated the fact that the train offers several compelling advantages over other means of transport: it is safer, less polluting, its infrastructures have a limited environmental impact, and finally it offers the highest density of transport (in terms of passengers and freight). For example the energy and social costs of rail are respectively 60% and 90% lower than by road transport.

The railways have realized the opportunity to play a leading role in the overall future transportation system, but the transformation of this potentiality into reality has required that the entire sector had a radical change and modernization.

To solve this problem, from the mid-90s the main European railway authorities, stimulated by the European Union have launched a massive development plan for facilities with significant investments in the Information and Communication Technology also with the aim to have a rail network with the same technical standards creating the interoperability and so knocking

down the limits due to different power supply and signaling systems.

In recent years (since 1998) in Italy, after a few isolated trials, a significant process for a new technologies introduction has began, mainly in the transportation safety management (Command Control and Interlocking Systems).

Regarding to the safety of train running, the Italian railway system history was limited to the management of spacing between trains and the availability of the track made in different ways.

For example

The Italian Railways Company (RFI) has therefore initiated, since 2000, a homogenization process of the protection level during the train running by introducing the System SCMT and the SSC systems that perform the Automatic Train Protection (ATP) providing the necessary safety conditions for trains minimizing risks due to human error.

These are discontinuous systems based on the transmission to the on-board computer of the information related to the optical signals along the line, through balises (SCMT) or Transponders (SSC), that permit to stop the train in case of driving error.

The overall development plan, including the introduction of high technology and equipment for railway systems, had the following goals for the rail system:

A. Improve safety levels, efficiency standards and reliability,
B. Offer new functionalities,
C. Introducing new criteria for maintenance (Senesi, 2007c),
D. Support a new work organization.

The innovative systems implemented (Senesi, 2009b) on the entire traditional network are:

1. ACS(Apparato Centrale Statico), a Computerize Interlocking System for the train stations management
2. SCC (Sistema Comando e Controllo), Command and Control System for a centralized over long rail distances management
3. SCMT (Sistema Controllo Marcia Terno) and SSC (Sistema Supporto alla Condotta) - systems for the Automatic Train Protection
4. GSM-R (GSM-Railway communications), a GSM standard by adding new features for railway operations.

At the same time, internationally, with the frontiers opening and the establishment of a Common Market, rail interoperability became fundamental and inevitable, to encourage goods and people movement.

Many problems had to be solved to reach a European railway standard: the extreme rail signaling systems diversity between countries (see Figure 1), even as the different infrastructure characteristics These limitations did not allow a free international rail traffic.

On border it is necessary, in fact, to change the engine (and eventually the whole rolling stock) and driver, to satisfy different operating rules related to national signaling systems (it is as if for a car you had a different gasoline and road signs).

Finally, after years of study, a technological system (ETCS, European Train Control System) for the European Rail Traffic Management had been reached.

The ERTMS is the union of ETCS plus the GSM-R. In that way the on-board and the trackside ETCS subsystems are connected by a GSM-R communication.

Until this moment the most important signallig systems in Europe were the LZB (Germany and Austria) and the TVM (France, Belgium, Great Britain). The first can be considered the old ETCS system as it permits a continuous transmission between train and central control system, the second instead is similar to an ATP system (like Italian system SCMT+BACC) and has therefore a discontinuous transmission of the information to the train. LZB and TVM can be considered the

parents of the ETCS even if they aren't interoperable systems.

As the ETCS deals with a wide range of operational requirements and its introduction on the entire network cannot be rapidly achieved, a gradual implementation has been planned compatible with existing systems.

To meet these needs it is possible to overlay the new European standard in national systems in order to permit a mixed traffic with contemporary movement of trains equipped both with ERTMS and not. Adopting the ERTMS is possible to reach the highest degree of sophistication, implementing not only the Automatic Train Control (ATC) which enables the protection and supervision of the train movements, but also obtaining the highest level of integration between the technological systems.

In Italy the ATC system has been installed on approximately 900 km line from Turin to Milan until Naples (with the exception of the Florence-Rome, already equipped with a "RS9 codici", the Italian signal repetition system with 9 codes).

Obviously the safety level achieved by these systems is so high that the possibility of an accident is almost zero as they are built according to the maximum rail safety international parameter (CENELEC Safety Integrity Level 4).

This has allowed the Italian railways to continue the gradual reduction in the number of accidents in recent years placing RFI at the top of the international security ranking.

Social Implications

Last year, an additional line, the High Speed one, has allowed us to increase traffic, to separate local/regional traffic with long distance trains giving great advantages to the regularity of the service.

The choice of a High Capacity High-speed line will also allow the rapid transport of goods among countries through international frontiers.

In the next future the High speed network expansion completing the Milan-Venezia with any other southern extension will surely further increase the social and economic benefits such as enhancing the already thriving port and industrial reality in some centers (e.g. Nola, Gioia Tauro, Bari) with the presence of hubs for the storage and handling of goods.

RFI LIFE-CYCLE ETCS SYSTEM

RFI follows CENELEC EN-50126 and EN-50128 standards for the development, assessment, product approval of products and electronic systems for railway safety (see Figure 2), by Disposals 32 and 39 in the year 2002 has defined the application rules (see Table 1). RFI has also defined the general operating procedures for implementation and for other additional configurations of ERTMS/ETCS Level 2 system regarding the HS/HC lines from the application until the putting in service.

Later, there has been the RFI PROCEDURE 009 that has permitted the application of the DISPOSALS 29/2002, 32/2002, 16/2003 (see Table 1).

Specific disposals have been adopted by the Technical Verification Commission concerning the putting in service of the ERTMS-ETCS system, in particular the signaling and spacing system.

RFI has therefore followed the process described in CENELEC 50126. After a first step for the "conception of the system ", RFI evaluated the target with the applying conditions, planning a delicate phase that involved the risk analysis (FTA-FMEA-FMECA-HR-Markov-Security) and produced by its competent structures, a Preliminary Hazard Analysis document (PHA), which has then been used as input by the Consortium of Suppliers (Saturno) for the System Hazards, for the Risk Analysis and for the issue of subsystems requirements specification.

About 250 hazards have been found and the 30% of these has been mitigated by solutions approved by the Ministry of Transport.

Figure 2. CENELEC 50126 Life-Cycle

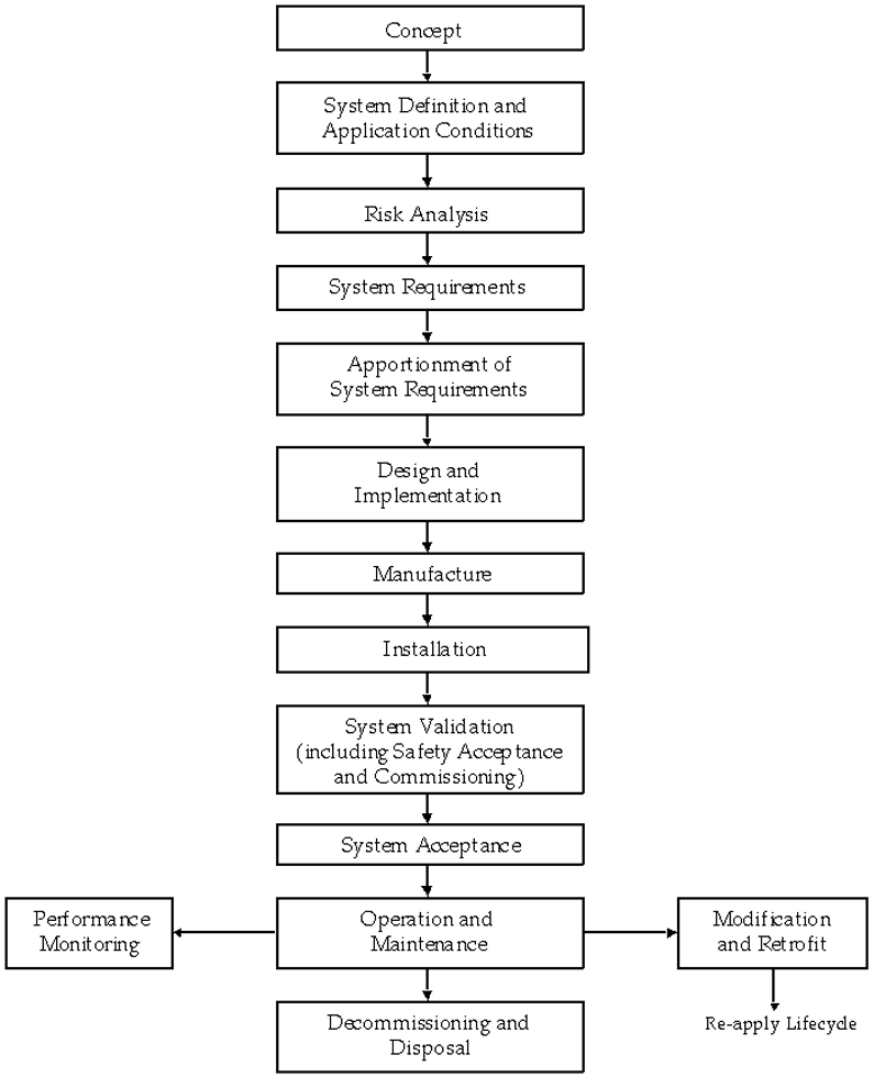
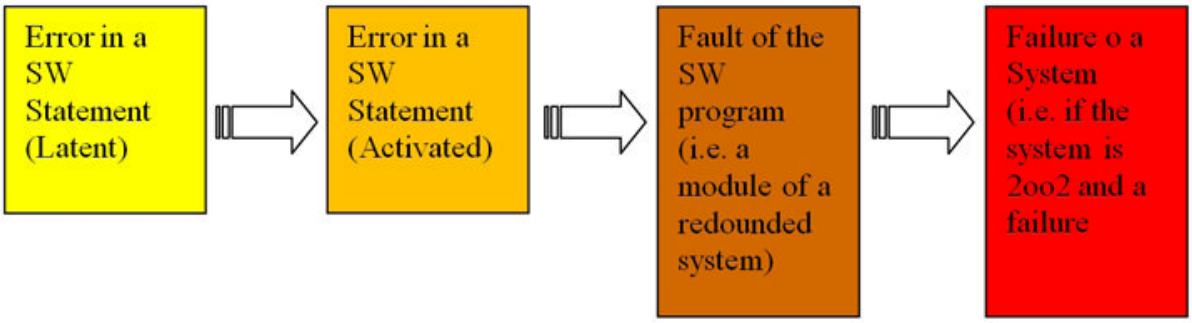


Table 1. RFI Disposals for the development and service of the railway signaling systems

	ARGUMENTS
DISPOSAL 29/2002	Developing and carrying out of technological products and systems for railway signaling. On-board and trackside subsystems are developed respecting the CENELEC's
DISPOSAL 32/2002	It defines the principles for the application of The CENELECs It has been integrated with the Disposal 52/2007 It defines the activities to do for the new products/systems (to develop, in to manufacture, to homologate and to be used in other systems)
DISPOSAL 16/2003	It disciplines the putting in service and the roles, the relations and the competences for each function and for each activity. It specifies and describes the function of each responsible (Engineering technologist, Regulator, Organizer, Functional Customer, Constructor, User, Maintenance, Local Technical Reference) It specifies and describes the necessary activities (Putting in Service, Projecting, Technical Assessment, Modifying of the existing, Periodical Technical checking, etc.)

Figure 3. Example of a possible sequence Error-Fault-Failure for a system base on software



The identified hazards were included in a hazard log that has been adopted during the project until the initial period of the commercial service.

The Suppliers Consortium (Saturno) finally issued two documents relating to the FTA and FMEA analysis of the system to finally reach the system document on the Hazard Failure Rate (HFR).

Hence on the basis of specific European UNISI, RFI defined General System Requirements (SRS Vol.1) by which the suppliers have drawn up the track-side and on-board subsystem specifications (corresponding to SRS Vol.2 and Vol.3).

The SRS Verifying activity formed part of the planned V&V and "Reverse Engineering" processes for the AV signalling systems and was carried out by the Suppliers Consortium (see Figure 3 and Table 2 for error propagation).

The requirements of the subsystems were then further subdivided and classified on lower hierarchies.

A dedicated RFI working group (ATC Project) has monitored the system development under continuous supervision and comparisons with suppliers.

A commission called "Plenipotentiaries", formed by Client (RFI), Contractors (Italferr) and Suppliers, made it possible to manage, directly and quickly, both technical and organizational issues.

For the development of every sub-system/ product, RFI has checked the SIL, evaluating the hardware components and Software. As it's not possible to give a numeric value to the last one, the supplier has been required to follow the process defined in the CENELEC's, the simplified scheme concerning the organization of the supplier and is represented in the Figure 4 I. This process has been thought to reduce systematic errors not otherwise quantifiable.

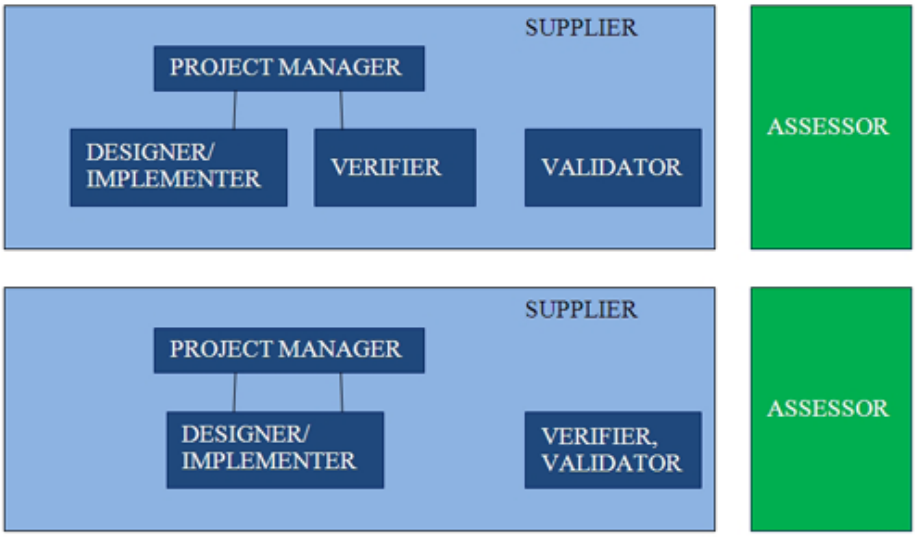
Instead of random hardware failures there have been made an estimate by mathematical probability, often using the Markovian theory, depending on:

- Hardware components failure rate (HR)
- Failure modes
- Mean detection and repair time.

Table 2. Definition for the sequence error-fault-failure-hazard

Error	**Error in the project that may lead to a Fault**	**If a SW latent error is activated it generates a fault**
Fault	anomaly condition that could lead to system failure (Failure)	A fault may cause a malfunction
Failure	A malfunction causes a deviation from the expected performance of the system	The failure creates Hazard situations
Hazard	Dangerous situation for people	No longer system protection for dangerous situations

Figure 4. SIL 3&4 Responsibility Process Organization



These evaluations have also led to logic system changes in order to achieve the SIL4 (see Figure 5) or to optimize the exercise regularity.

Italferr, with the role of General Supervision (practically the works manager), has monitored the manufacturing and installation process.

Therefore, the very important final step, for the validation and acceptance of security systems / products, has been carefully followed.

Safety Cases (for new products/systems) and Security Technical Report (for old products/ systems but already approved to be installed on 25KV electric power rails) were issued according to the hierarchy of systems / products in agreement with EN 50129. These documents covered step by step the whole process starting from the components until their integration.

With reference to EN-50128, both the supplier and RFI have therefore planned a series of tests to verify the correct implementation of each requirement requested by the product. RFI in particular, in addition to exploiting its ETCS laboratories (currently unique in their completeness and potential) for test scenarios, has developed formal languages analysis and verifications of the ETCS system.

Obviously, it is simpler and easier to repeat in a laboratory some situations that otherwise are difficult to achieve in the field, such as performance tests with 30 trains simultaneously connected to RBC.

Field trials, to complete the test process, have been carried out by RFI / TRENITALIA / Suppliers mixed staff, following incremental proves by a not homologated single train until a train chase, increasing speed gradually and with small distances between trains, of course adopting a strict safety regulation approved by the Ministry of Transport.

Figure 5. Safety Integrity Level values for Hardware products/Systems

Safety Integrity Level (SIL)	Probability of a Dangerous Failure (Per Hour in Continuous Mode Operation)
SIL 1	$\geq 10^{-6}$ to $< 10^{-5}$
SIL 2	$\geq 10^{-7}$ to $< 10^{-6}$
SIL 3	$\geq 10^{-8}$ to $< 10^{-7}$
SIL 4	$\geq 10^{-9}$ to $< 10^{-8}$

Technical verification committees for each subsystem were constituted and ETCS was one of the most innovative evaluation procedures. In addition, the ETCS system contains much of the data infrastructure and then it completes the final configuration process (Data Track condition, gradient profiles on weapons, telecommunications, management of the street, etc....). Moreover the ETCS system is strongly influenced by the telecommunications system.

The ETCS Level 2 Technical Verifying Commission has essentially provided the evaluations procedures of the first specific application of this system, as mentioned above, using laboratory and on field tests for the assessment of the proper implementation and functional compliance.

The findings were formalized in an appropriate documentation including not-compliance requirement (binding or not for the putting into service), to be closed during the pre-exercise (2/3 months normal exercise without passengers) before the commercial service, to evaluate the entire system and its service regularity.

RFI engineers have also supervised the operators training by enabling courses. The process just described was then reapplied to any subsequent system modification, according to EN- 50129, regarding the security lyfe-cycle.

Concerning the RBC, a functionality verifying process has been adopted for all possible features implemented, following the ETCS test specification provided by the Supplier's Consortium, using state-diagrams, checked through a reverse engineering process using the RBC data log files. For this activity RFI was supported by the Power System Group of the Salerno University. This process has analyzed the RBC log-files verifying their correspondence with the test plan output (Senesi, 2006a).

Not the least a GSM-R interference resistance analysis was also carried out through a comparison between measured data and potential interference sources (provided by the Ministry of Telecommunications)along the HS/HC trackside (Giugno, 2008). In this perspective, the experimental project GRIDES (Senesi, 2009c) (Baldini, 2010) for the GSM-R interference detection (accidental or malicious) was developed with the economic support from the European Commission and the technical assistance of Intecs Company and Pisa University.

The parallel development activities by suppliers and assessments by RFI have allowed a continuous retrofit of the onboard and trackside subsystems specifications (the last one with greater freedom degrees concerning constraints imposed by TSI), anticipating some change requests to the UNISIG specification (Unisig, 2002).

RFI TESTING EXPERIENCE AND FORMAL LANGUAGES

RFI has always paid attention to the test campaign, being a fundamental activity before the putting in service. Until now tests were typically based on real train runs according to a detailed program covering all the required functionality.

The problems were the high number of test runs and the correctness of the tests.

For the first problem RFI developed, together with the main suppliers Ansaldo, Alstom and Sirti an ETCS Laboratory with real target machines. For the evaluation of the test planning of the suppliers, RFI has developed a formal languages analysis with the Power System Group of Salerno University.

RFI ETCS Team has evaluated the Formal methods as the best solution for its test and specification activities, in fact the increasing complexity of the specification for railways application and the high number of supplier advice to use this approach to solve problems like:

- Misinterpretation requirements
- Correct but ambiguous interpretation of the requirements

- Possibility to use formal language tool for an automatic evaluation of the specification
- Possibility to use formal language tool to find new requirements for a better specifying (in particular for the not nominal behavior of the state-machine) (Antoni, 2008)(Amendola, 2003)(Hyun-Jeong Jo, 2008)(Senesi, 2006a).

It's not a coincidence that CENELEC 50128 strongly advices to use formal languages.

RFI experience on the requirements management, advices to provide a Boolean translation of the specification to give a simpler and clearer understanding. This suggestion is necessary as RFI has noticed that sometimes different suppliers give a correct but different interpretation for a requirement causing problems for the interoperability.

They introduce a high level of specification abstraction, they are supported by computer tools that permit to verify the compliance of the model with the specifications, the automatic verifying of inconsistencies such as logical conflicts, indeterminism, inaccessibility of states, and they check the accessibility to the states operating under specific conditions according with the requirements.

Moreover, the availability of a signaling system model based on a formal representation has permitted to develop a process based on the verifying of the system using log files generated during a real journey or in a test one generated by the RFI target machines in the ETCS Laboratories.

Amongst the formal methods used in literature, RFI has selected State-charts, one of the most widespread used languages for graphical formal specifications and very similar to the Flow-Charts presented in UNISIG Specification (Unisig, 2002). State-charts have also been chosen as their understanding is very intuitive and therefore easy to manage by everyone. In fact, other works on formal languages are often so complicated to be rejected or ignored by most of the people.

The best Tool for the formal verification based on State-charts, according to a "Milan Polytechnic" evaluation was founded in the tool "Statemate Magnum". This instrument had already been adopted for many automotive, railway and avionic applications.

In fact, the problem for the diffusion of this models is that the software for formal specifying is not user friendly so this tools can be used/evaluated only by people with a certain expertise. For example ERA has commissioned a process for the formal requirements specification, but the difficulty is that these tools use a mix of commercial and dedicated software requiring strong skills by users.

In the last 5 years, Statemate Magnum and its tools have been evaluated by RFI, not only as excellent instruments for formal specifications but also for their usability and intelligibility.

Statechart is expressed schematically through a state-based visual language, and therefore easy to understand. This graphic approach allows a system representation with different degrees of complexity. Moreover Statemate, which supports both formal specification and formal verification based on the Statechart, offers a "Certifier" able to verify visually the correctness and the evolution of the model through simulation of its graphic representation.

The automatic verification by this "Certifier" permits to recover the time spent for the formal developing of the statechart specification.

RFI has also been promoting the introduction of methods and systems able to support the transparent management of ETCS specifications, allowing both identification and correction of errors and ambiguities, and early verification of modifications and functional extensions. Statecharts are also the easiest and most comprehensible way to represent a specification without using natural speech language, it could be used like a universal specification language.

RFI has adopted formal languages for test specification since the 2004, for the putting in service of Roma-Naples ERTMS line and all the following High Speed lines. The process has been

developed to assess the ETCS Trackside Subsystem, permitting to evaluate if the real RBC was compliant to the test specification, and of course to evaluate through formal languages the test specification too.

In the development of a formal language test model, great attention was paid to its usability by the technicians of RFI and, more generally, by operators of the railway area: in compliance with this requirement, a "train-centric" model has been developed, where all the actions are related to the train. This approach guarantees a high correlation between the sequence of actions that characterize the evolution of the logic of the system with what happens physically along the line, offering a model of immediate interpretation by the "Domain Experts" (senesi, 2007a)(Senesi, 2006b)(Piccolo, 2009).

A test formal model can be defined starting from the UNISIG specification coded in transactions or states. The result of this process is a State-chart diagram organized in several functional scenarios one each other related (e.g. Connection, Start of Mission, End of Mission, L0/L2 transition, etc.). The advantage of using functional scenarios is to reduce the complexity of the entire state-chart model and to permit an easier comprehension and management by every user (Senesi, 2006a) (Senesi, 2007a).

A state-chart test specification can be modeled in Statemate in order to verify its completeness and correctness. To do this, the tools Model Checker and Model Certifier can be used to perform an automatic analysis of the model.

After a model verification by Statemate tools, the model can be explored using RBC log files generated at the RFI laboratories in Rome or obtained by real train runs, allowing a comparison between the logical model and the real machine. To quickly manage the log files a specific software has been implemented with the Support of The Power System Group of the Salerno University This software permits to convert the html RBC log file into a ASCII file Statemate compliant. These tests have supported the technicians of RFI in the evaluation of the behavior of the real system implemented, highlighting which functions of the model and of the specifications are actually used in real or test trip. The results of this analysis has to demonstrate that the test model is compliance with the specification and consequently the ERTMS Subsystems (RBC and EVC) are compliance to specifications too.

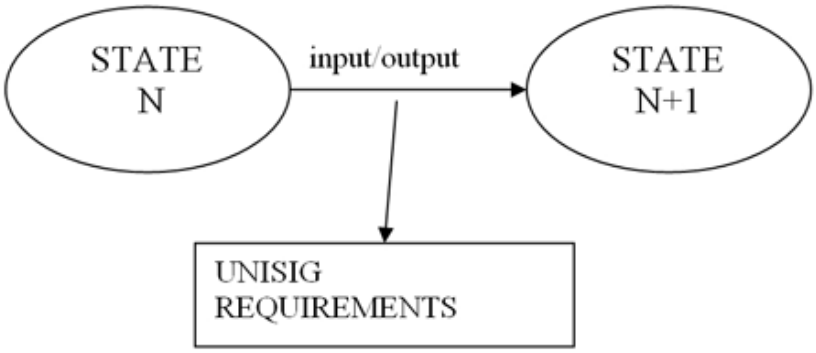
According to the experience RFI has gained the Test Implementation Phases for every new similar activity and the criteria for the work are:

1. The personnel dedicated to the test modeling activities are chosen according to the experiences and competencies on the ERTMS system. The tests are managed according to ISO9000 series quality management standards and the ISO 17025 assessment process.
2. Every test ha s a scheduled sequence of input operations and output results of the system. The test is described both in statecharts and in natural language.

Every transition (Single Test Case) on the logical model represented in the Figure 6 between states must be linked to one or more UNISG requirements. A Test can be built as a sequence of single test cases.

3. RFI uses Statemate Tools for a first logical verifying of the model correctness. The model can be evaluated with the Statemate tool "Model Checker" so that it's easy to find eventually errors like indeterminism, no reachable states, unwanted behaviors.
4. RFI uses its ERTMS laboratories at the "Istituto Sperimentale" in Rome for test model correctness verification by RBC and EVC Log Files. This phase consists in the conversion of several different log files of the ERTMS Subsystems to only one Meta Log file and then, knowing the statechart

Figure 6. Example of transition between states



variables and rules it's possible to obtain a test file Statemate compliant (Figure 8), so that the model can be explored and verified and we can reproduce the logical sequence of the events recorded during the test.

5. If a change of UNISIG occurs it is easy with a statechart model to find the test cases linked to each requirement and so it's possible to correct the test cases changing their description and outputs. The management of the modified requirements is a practice that RFI well knows and has adopted during the last years for the verification of new system releases.

This procedure (Figure 7), adopted by RFI for the test specifying of the Italian High Speed Lines and for the Modeling of the Interconnection Functionalities between the signaling on the historical lines and ETCS Level2 on the high speed network, has permitted to find errors due to human factor and to reduce the time necessary for the test result verifying and for the test specification rearrangement due to Unisig's modifying.

All the ERTMS lines (Table 3) have been tested by formal method by RFI.

Next Step is in progress and regards the modeling of the all ETCS Trackside Specification.

ETCS LABORATORY

Since 2007 RFI with the support of Ansaldo, Alstom and SIRTI has developed in Rome and integrated ETCS+GSMR laboratory (Figure 9) for the integration testing between RBCs and EVCs.

The Main features of the laboratories are:

- To do interoperability test
- To build off-line nominal and degraded scenarios
- To import RBC log file pertaining to real time scenarios and "replay" them with the aim to analyze functionalities and faults
- To meaningfully reduce the number of test on site necessary to launch a new software configuration in commercial service
- Possibility to test a number of new track-side configurations over real homologated on-board sub-systems.

The architecture of the Laboratory is based on:

- Real RBC sub-system with interface to GSM-R network and IXL (one subsystem for each supplier)
- Simulation of IXL sub-system (e.g. Rome-Naples) (one subsystem for each supplier).

Figure 7. RFI management process for log file analysis with formal languages tool

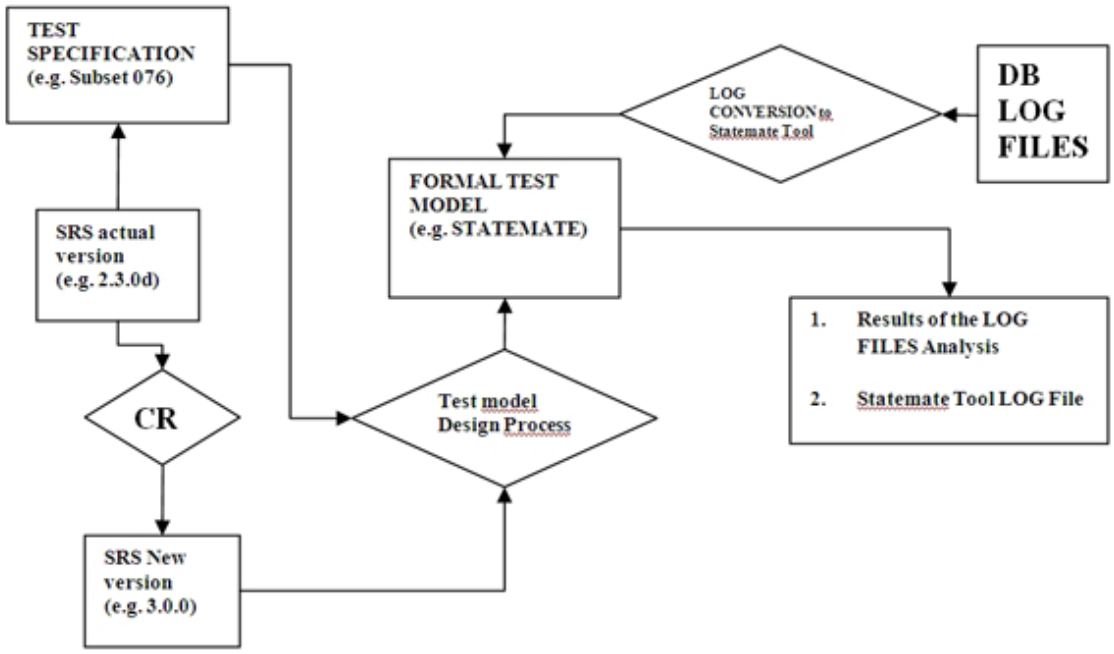


Figure 8. RBC log files merging and fusion process

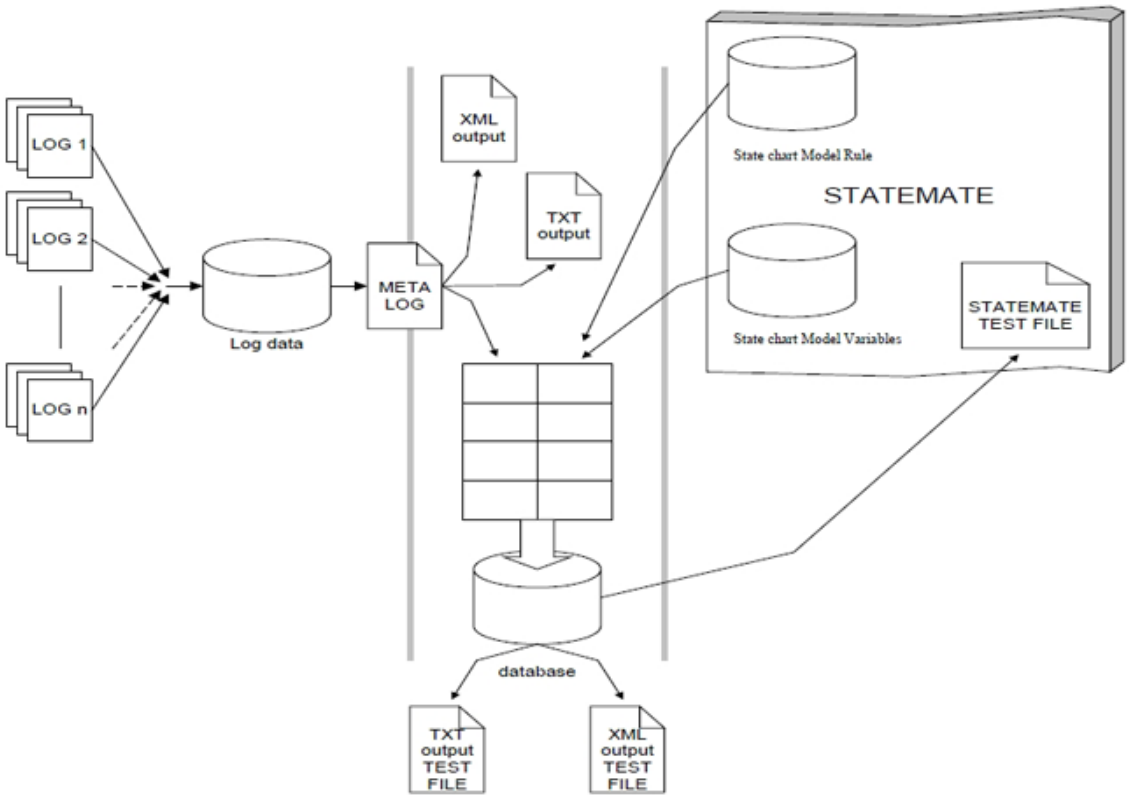
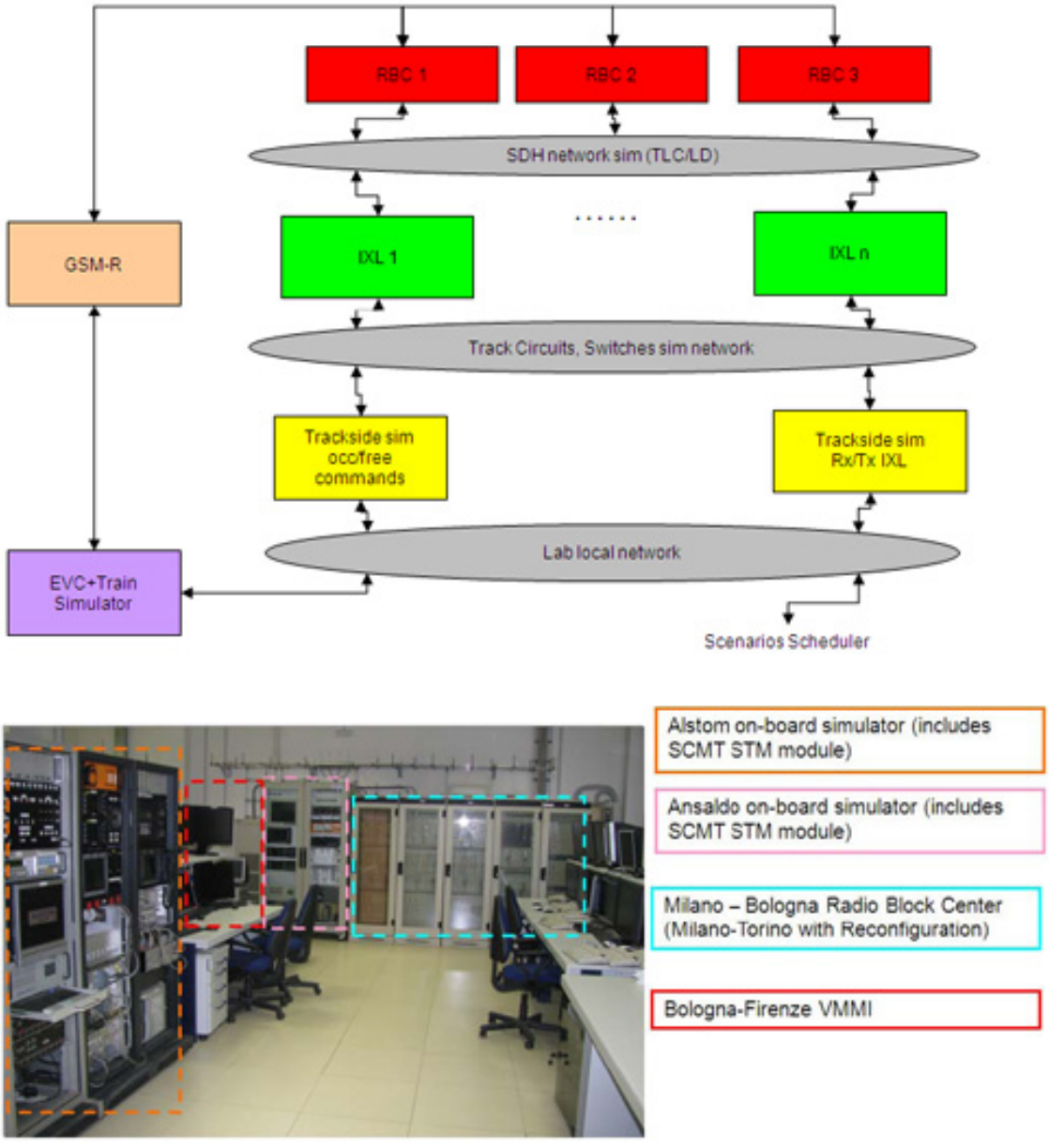


Table 3. ERTMS/ETCS Level 2 HS/HC Italian Lines

Line	Commercial Service
Rome-Naples	2005
Torino-Novara	2006
Milano-Bologna	2008
Bologna-Firenze	2009

Figure 9. Scheme and Overview of the ETCS Laboratories at "Istituto Sperimentale" of RFI



- Sub-system supervising and remote control with DCO/DM trackside operator interface.
- Real EVC sub-system linked to a Train simulator with interfaces to GSM-R network (Mobile terminal and Euroradio module) (one subsystem for each supplier)
- EVC Simulator subsystem to test undesired scenarios.

RFI has installed RBC's, EVC's and GSMR in an area where it is possible to reproduce test scenario on real target machines. Practically it's possible to connect through the GSM-R the EVC to the RBC (but it's also possible to use a LAN connection). So it has been possible to do test with products both of some supplier and of different suppliers; for example on the same RBC we have connected a EVC of Ansaldo and another EVC of Alstom verifying the interoperability and the train spacing.

It's easier to do performance tests like the simultaneous connection of 30 trains with a RBC.

The only simulated thing is the track occupancy due to a real train, but this is easily done with a software that evaluates the train position and send to the Interlocking system the information about the track circuit state.

There is of course a panel control like that installed in the Control Stations for HS lines, so that every test can be evaluated first on the monitor and then analyzing the Log File of The RBC and EVC. At the end we can say that there is no difference between the ETCS laboratory and the ETCS line in Italy because they use the same technology and have the same configuration.

The testing activity is fundamental to release a product/system compliant with the safety required, so the process adopted using state-charts permits to have a test scenario easily and quickly reproduced in the laboratory.

Consequently the ETCS subsystems Log files are available for an analysis through the Formal Language Tool after their conversion into a Statemate Compliant Log File

FINITE MARKOV CHAINS ANALYSIS

Basis on the concept of state model, the logic behaviors of some systems were analyzed, evaluating, by Markov theory, the probabilities of hazard situation or unavailability. These have been used to choose each time the most suitable solutions or to modify the system logic to respect the Cenelec SIL4 (Senesi, 2006c).

Examples are the evaluation of the necessary of a redounded RBC system, and the choice of a different logic implementation, compared to UNISIG, on the hot axles box detection system (Senesi, 2010). In the latter case, UNISIG provided a four balises information point and to stop the train in case a single balise was not detected.

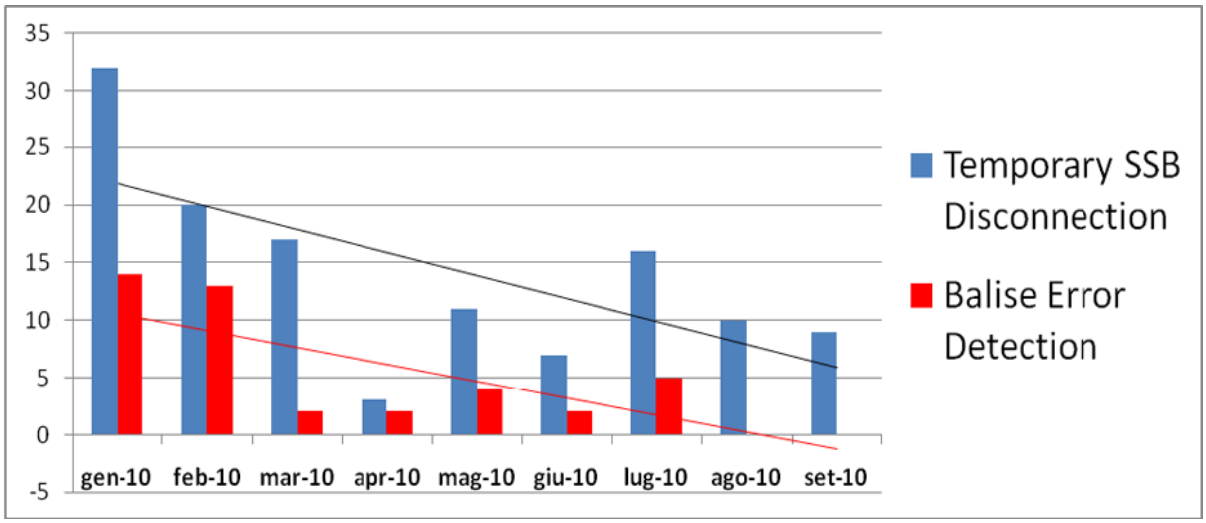
This should have been, compared with a total security, a problem for the regularity, because it is likely that sooner or later a train misses the correct reading of a balise.

The probabilistic assessment has permitted to choose the best solution (placing two consecutive balises groups, only 2 balises each one) for hot axle alarm transmission to the train, and at the same time guarantee the train stop only if they none of this balises group is completely detected by the train. Therefore, security is maintained as in the original UNISIG specification, without frequent stops of the train and then increasing the service regularity.

OPERATION AND MAINTENANCE

RFI and Suppliers after the putting in service of a new HS ETCS line have always played much attention to the service regularity; in fact daily reports and graphics are continuously available and, if necessary, dedicated staff for each subsystem are involved. For example, in Bologna, at the Central Control Station for the lines Milano-Bologna-Firenze, there are specialized rooms for the supervising of the systems. So each Central Control Station can analyze in real-time and off-

Figure 10. Decreasing trend of delay causes on Milano-Bologna due to transient faults



line all the information through Log files of the subsystems. RFI has also developed an independent sniffer (named SMAV) for the transmission between RBC and EVC and between RBC and IXL system; it's placed in Rome Central Station and permits to decode the euroradio messages sent and received by RFI GSM-R network (Senesi, 2009a).

These analyses permit to improve the traffic regularity observing problems, their frequencies and places where they are. It's possible to see the result in the Figure 10.

It's important to say that diagnostic dedicated trains named Archimede and Y1 are able (at 300km/h speed) to capture and analyze all the line parameters (ETCS, GSMR, track gauge, pantograph interaction,..). The RFI Group responsible for mobile diagnostic is improving the HS line trains following the certification experience on the traditional lines (Favo, 2011).

CONCLUSION

Since the putting in service of the HS / HC lines RFI has been monitoring the service regularity and the products availability, both in contractual terms and for the continuous improvement of system performances, taking action where possible by better products and updates to the logic of ERTMS subsystems (RBC, EVC, GSM-R).

System changes and upgrades have been proposed both by RFI and by suppliers.

After the release of technical documents and the verification and validation plan, RFI evaluates the interventions to be carried out and authorizes the changes.

RFI has compared ERTMS with SCMT+BACC(the actual best signaling system on the historical line) and, following a CENELEC Common Safety method, RFI has developed a Hazard Analysis to apply ETCS on the historical lines focusing the attention on the safety requirements according to Italian railways regulation.

The Formal languages and ETCS Laboratories have permitted to increase the safety of the assessment and to reduce the time necessary for each prove.

The experience gained will be useful in the future for new similar lines like the Milano-Venezia line and frontiers (provided by the European transport plan).

Concerning the social implications, last year, an additional line, the High Speed one, has allowed us to increase traffic, to separate local/

regional traffic with long distance trains giving great advantages to the regularity of the service.

The choice of a High Capacity High-speed line will also allow the rapid transport of goods among countries through international frontiers.

In next future High speed network completing on the Milan-Venetia and any other southern extension will only further increase the social and economic benefits such as enhancing the already thriving port and industrial reality in some centers (e.g. Nola Salerno, Gioia Tauro, Bari) with the constitution of hubs for the storage and management of goods.

REFERENCES

Amendola, A. M., di Maio, R., Iacobuzio, M. L., Poli, F., & Scalabrini, F. (2003). Lessons learned in designing and evaluating railway control systems, *Proc. of. Ninth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems*, (pp. 355 – 358) 1-3 Oct. 2003

Antoni, M., & Ammad, N. (2008). Formal validation method and tools for French computerized railway interlocking systems, *4th IET International Conference on Railway Condition Monitoring*, 1-10

Baldini, F., Luise, P., Bagagli, R., Marcoccio, M., Senesi. M, (2010). The High Speed Railway infrastructure in Europe and its dependency on GSM-R wireless communication, *International Journal of Critical Infrastructure Protection*, 2010.

European Commission. (2006). *Sustainable Surface Transport Research Technological Development and Integration*.

European Commission. (2010). *White Paper - European transport policy for 2010: time to decide*, ISBN 92-894-0341-1

Favo. F., Alterisio G., & Illibato G. (2011). La certificazione in esercizio delle linee SCMT, *Tecnica Professionale [Professional Techniques]*, May 2011

Giugno L., Luise M., Bagagli E., Giannini M., Senesi F., Malangone R., & Caronti D. (2008). Valutazione dei fattori di rischio indisponibilità nell'uso della rete radio GSM-R per applicazioni ferroviarie italiane ad alta velocità/alta capacità (AV/AC) [Assessment of risk factors is unavailable in the use of the radio network for GSM-R applications Italian railway high speed / high capacity], *Ingegneria Ferroviaria [Railway Engineering],* January 2008

Jo, H., Hwang, J., & Yoon, Y. (2008). Applying formal method to train distance control system by combining ZED and Statechart, Proc. *of International Conference on Control Automation and Systems,* (pp. 896 – 900) 14-17 Oct. 2008.

Obama, B. (2008). *Strengthening Americans Transportation Infrastructure*. Retrieved from. http://www.barackobama.com/pdf/issues/FactSheetTransportation.pdf

Piccolo, A., Senesi, F., Galdi, V., & Malangone, R. (2009). Use of Formal Language to represent the ERTMS/ETCS system requirements Specification of the Interconnection L0/L2 - *International Conference on Model and Technologies for intelligent transportation systems*, Università La Sapienza in Roma 22-23 June 2009.

Senesi F. (2009b). Sistemi di protezione e controllo della marcia dei treni (ERTMS, SCMT, SSC). applicazioni e sviluppo per la rete ferroviaria nazionale italiana [Train running protection control systems (ETCS, SCMT, SSC). application and development for the italian railway system], *Ingegneria Ferroviaria [Railway Engineering]*, April 2009

Senesi, F., Bonafè, G., Geraci, S., Frandi, M., Filippini, N. & Malangone, R.(2008). Getting ERTMS into service quicker ETR 500 test train, *Railway Gazette International*, December 2008.

Senesi, F., Filippini, N. & Malangone, R. (2009a) RFI – SAFETY & SIGNALLING ON FLORENCE-BOLOGNA, *Eurailmag 03*

Senesi, F., & Malangone, R. (2007a). Formal method analysis and evaluation of ERTMS-TEST specification for the Italian high speed railway, *Proc. of Formal 6th International Symposium on Methods for automation and safety in railway and automotive systems - FORMS/FORMAT 2007*, 25-26 January 2007.

Senesi F. & Malangone R.(2010). Ram analysis of the Radio Block Center System for Italian ERTMS lines, *Ingegneria Ferrviaria [Railway Engineering]*, April 2010

Senesi, F., Malangone, R., & Petaccia, G. (2006c). Utilizzo della distanza obiettivo come seconda catena di appuntamento nella Logica SCMT, Ingegneria Ferroviaria, ed. settembre 2006.

Senesi F., Malangone R., Piccolo A., & Galdi V. (2006a). Utilizzo di linguaggi formali per la analisi e la valutazione delle specifiche di test del sistema ERTMS della rete italiana ad Alta Velocità, [Using formal languages for analysis and evaluation of the test specifications of the ERTMS on the Italian network to High Speed *]Ingegneria Ferroviaria*, [Railway Engineering] December 2006.

Senesi F., Malangone R., Piccolo A., & Galdi V. (2006b). Utilizzo di linguaggi formali per la analisi e la valutazione delle specifiche di test del sistema ERTMS della rete italiana ad Alta Velocità, [Using formal languages for analysis and evaluation of the test specifications of the ERTMS on the Italian network to High Speed] *Ingegneria Ferroviaria* [*Railway Engineering*], December 2006.

Senesi F., Malangone R., Rossi C., & Torassa M. (2007c). Le Apparecchiature del Sistema Controllo Marcia Treni [The Equipment Control System Run Trains], *Tecnica Professionale* [Professional Techniques], October 2007

Senesi, F., & Marzilli, E. (2007b). *European Train Control System - Development and Implementation in Italy* Retrieved from: http://www.etcsbook.com.

Senesi, M. (2009c). Il progetto GRIDES per la disponibilità della rete GSM-R, [The project GRIDES for the availability of GSM-R] *Tecnica Professionale* [Profession Techniques], September 2009

UNISIG. (2002), *ERTMS/ETCS – Class 1 – System Requirement Specification, 026*,(2.2.2)

KEY TERMS AND DEFINITIONS

ATC (ATP). Automatic Train Control (Protection)

CR: Change Request

ERA: European Railway Agency

ERTMS: European Railway Traffic Management System

ETCS: European Train Control System

EVC: European Vital Computer

HR: Hazard Rate

HS/HC: High Speed/High Capacity

HW: Hardware

IXL: Interlocking

RFI: Rete Ferroviaria Italiana

RBC: Radio Block Center

SCMT: Sistema Controllo Marcia Treno (Train Movement Control System)

SSC: Sistema Supporto Condotta (Driving Support System)

SRS: System Requirement Specification

SW: Software

Chapter 19
Adoption of Low-Cost Rail Level Crossing Warning Devices:
An Australian Case Study

Christian Wullems
Cooperative Research Centre for Rail Innovation, &Centre for Accident Research and Road Safety – Queensland (CARRS-Q), Australia

George Nikandros
Australian Safety Critical Systems Association, Australia

ABSTRACT

The objective of this chapter is to provide rail practitioners with a practical approach for determining safety requirements of low-cost level crossing warning devices (LCLCWDs) on an Australian railway by way of a case study. LCLCWDs, in theory, allow railway operators to improve the safety of passively controlled crossing by upgrading a larger number of level crossings with the same budget that would otherwise be used to upgrade these using the conventional active level crossing control technologies, e.g. track circuit initiated flashing light systems. The chapter discusses the experience and obstacles of adopting LCLCWDs in Australia, and demonstrates how the risk-based approach may be used to make the case for LCLCWDs.

INTRODUCTION

Australia has a large number of passively controlled road rail level crossings on railways. Individually such crossings are relatively low risk, however collectively these passively controlled crossings represent a significant safety issue for Australia. Australia is not unique; there are numerous other countries that have large numbers of passively controlled crossings. The high cost of the conventional active level crossing, e.g. track circuit initiated flashing light control technology, is a significant barrier to upgrading these passively controlled level crossings; the use of conventional high safety integrity technologies

DOI: 10.4018/978-1-4666-1643-1.ch019

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1. Level crossing with passive controls in Queensland



means that only a limited number of crossings can be upgraded each year.

The rail industry, both in Australia and internationally, has been trialling a number of low-cost level crossing warning devices (LCLCWDs). These devices use alternative, less-reliable lower integrity lower cost technologies, such that installation could be as low as 25% of the cost of the conventional technologies. These low-cost systems are not intended to provide a low-cost alternative to conventional systems on high exposure crossings, rather to provide active protection to low exposure crossings on low traffic train lines with few passenger services. An example of a level crossing with passive controls suitable for treatment with LCLCWDs is illustrated in Figure 1.

The basis for the argument for the use LCLCWDs is that for a given investment, more level crossings can be treated, therefore gaining much greater safety benefit for the same investment to treat one level crossing using conventional technologies.

In Australia, the current approach for the improvement of level crossing safety involves the incremental upgrade of passively controlled level crossings with active controls, subject to available funding. The prioritization of sites for upgrade is determined using a risk assessment model such as ALCAM (Australian Level Crossing Assessment Model) (Department of Transport NSW, 2010). ALCAM ranks crossings using a consistent basis according to a detailed level of comparable risk, exposure and consequence. The accepted use of models like ALCAM is some recognition by transport safety regulators that it is not always practicable to provide the most effective safety treatment possible; there is acceptance, albeit reluctant acceptance, of crossings with passive controls despite active control treatment technologies existing.

There are several issues that have hindered the adoption of LCLCWDs in Australia. This chapter discusses the key issues including reliability and the legal issues associated with reduced reliability. An Australian case study is provided, based on a high-level risk assessment and cost-benefit analysis.

BACKGROUND

This section provides background information on low-cost level crossing warning devices (LCLCWDs) and reliability issues; human factors issues that influence the effectiveness of such devices in improving safety; and obstacles that are impeding the adoption of LCLCWDs in Australia.

Low-Cost Level Crossing Warning Devices and Reliability Issues

Low-cost level crossing warning devices (LCLCWDs) are characterized by the use of alternative technologies for train detection (e.g. radar, magnetic induction, Global Positioning System, etc.) and connectivity (e.g. wireless communications); and often rely on the use of solar power. The target environment for their deployment would be a low risk crossing, e.g. crossings on a single-track line with relatively low vehicle and rail traffic and few passenger train services if any.

LCLCWDs are generally not considered to have the same fail-safe and safety integrity characteristics as the conventional technologies. Of particular concern are potentially increased rates of wrong-side failure. The rate of wrong-side failure (i.e. a failure that prevents the issue of a timely warning of an approaching train to the road user e.g. a train detection failure or a failure which prevents the road warning lights from flashing) needs to be acceptable; needs to be at a level commensurate with the risk.

Conventional level crossing warning devices are designed with high safety integrity train detection and control systems. Train detection systems are often based on technologies such as track circuits, axle counters or grade crossing predictors (GCP) that are of high integrity and usually would meet a high Safety Integrity Level (SIL) rating.

Safety integrity levels (SILs) are discrete levels that have an assigned dangerous failure (i.e. wrong-side failure) probability. For systematic failures, these levels are associated with a group of methods and tools, which if applied, should provide the evidence and confidence that system meets the SIL target. Safety integrity is specified as one of four discrete levels: Level 4 being the highest, Level 1 being the lowest. Level 0 is used to indicate that there are no safety requirements. Table 1 details the SILs for continuous mode operation.

Other than train detection and control system components, little is intrinsically fail-safe. For example, backup power, although redundant, is not fail-safe. If the crossing loses power, there is a window of time (assuming backup power does not fail) for intervention before the crossing fails in a dangerous manner. Warning signals to the road user are typically redundant, but can fail in such a way that no signal is emitted. The control systems typically include a supervisory function that monitors the health of the various subsystems including train detection and warning signals, providing local and remote fault monitoring. Remote diagnostics can facilitate signalling of the level crossing system's health state to train dispatchers, such that timely intervention can be made in order to mitigate the consequences of failures.

If it can be demonstrated that safety risks are adequately reduced with a lower integrity technology solution, it may then be possible to make an argument for the deployment of LCLCWDs using so far as is reasonably practicable as the basis.

Table 1. Safety integrity levels (Standards Australia, 1999)

Safety Integrity Level	High demand or continuous mode of operation (Probability of a dangerous failure per hour)
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

Human Factors Aspects of Availability

Human performance is a key aspect of the safety performance of any level crossing protection system. LCLCWDs are assumed to be less available than high-integrity level crossing devices due to the use of lower cost, lower safety integrity technology. As such, human factors issues relating to decreased availability through right-side failures (false positives) need to be carefully considered. A right-side failure is a failure condition that results in the system entering a safer state than had the failure not been present (i.e. in the case of level crossings, this means activating the crossing controls as though a train was approaching).

The perception of reliability of level crossing warning devices has been demonstrated to affect road user compliance in studies such as the simulation study conducted by the American Federal Railway Administration (FRA) and sponsored the John A. Volpe National Transportation Systems Centre. This study examined the effects of active warning reliability on motorist behaviour at rail level crossings (Gil, Multer, & Yeh, 2007). The effects of false positives (i.e. activation of the warning when no train is approaching) and false negatives (i.e. missed train detection) on motorist behaviour were examined in two experiments, which concluded that improving motorist's perception of signal reliability might improve compliance.

In the target low-exposure crossing environment of LCLCWDs, the risk of increased levels of right-side failure may be deemed tolerable. However, frequent right-side failures i.e. cry wolf type failures, could have a negative impact on road user behaviour, potentially reducing the effectiveness of the safety improvement.

Common failure modes for Australian actively controlled level crossings are as follows:

- Flashing lights of the RX5 assembly (cross-arm assemblies consisting of two lights for unidirectional or four lights for bidirectional) will flash and boom gates will lower in a known failure state for level crossings with active protection consisting of flashing lights and boom gates; and
- Flashing lights of the RX5 assembly will flash for level crossings with active protection consisting of flashing lights only.

An issue with the design of actively controlled level crossings with RX5 flashing light assemblies in Australia is that the expression of failure (right-side) is identical to the train approach warning, resulting in ambiguous communication of crossing state to the road user. Road users, when confronted with a train approach warning and no train approaching (right-side failure condition), may find themselves in a confusing situation in which they are unsure how to behave or what their responsibilities are. While there is an inherent safety benefit if a road user finds the warning device in a state of right-side failure (in that the road user is likely to stop due to the flashing lights), frequent and prolonged right-side failure may increase the likelihood that road users will engage in risky behaviour. Research is needed to quantify these effects in order to support the specification of high-level availability targets.

These issues, while not in the scope of this case study, are important aspects that should be considered in making the case for the adoption of LCLCWDs.

Obstacles for the Adoption of Low-Cost Level Crossing Warning Devices in Australia

This section will briefly describe some of the key legal issues that are hindering the adoption of LCLCWDs in Australia. These issues are mostly related to reliability aspects of alternative tech-

nologies used in LCLCWDs and the legal liability consequences that may arise due to their use.

The responsibility for level crossings is typically placed with rail infrastructure managers. Concerns have been raised as to whether rail infrastructure managers would be liable for injuries or fatalities occurring at level crossings protected with LCLCWDs with lower safety integrity and lower reliability technology. This is particularly relevant where an incident is determined to have been caused by a failure of the technology, because inherently safer technology was available at the time.

Railway practitioners have recognized the issue of Tort liability as determining factor for the acceptance of LCLCWDs, and a principal motive for which the United States and other countries have not adopted LCLCWDs to date. Roop et al (Roop, Roco, Olson, & Zimmer, 2005) state that:

"…if railroads (or any entity taking responsibility for risk at crossings) install equipment at a grade crossing that is in any way inferior to current technology – independent of cost – then liability will likely be maintained because the responsible entity knows that the alternative system is not as good as an existing system – an act of commission."

Roop et al (Roop, et al., 2005) further note that in the United States, a necessary precursor to change and eventual acceptance of new technology is Tort liability reform followed by a change in the apportionment of risk and liability. The new technology, however, would need to demonstrate an overall enhancement of safety.

In Australia, there have been several attempts to investigate and address the legal issues associated with the LCLCWDs. In 2006, Jordan (Jordan, 2006) noted that legal advice was obtained before the trail of LCLCWDs in the state of Victoria. The legal advice was of the view that the technology had to be well tested, subjected to rigorous risk assessments, and applied in a professional manner. It was concluded that if this were done, a Court of Law would most likely have little reason to find against the LCLCWD, all other matters being equal.

The Victorian Railway Crossing Safety Steering Committee subsequently sought advice from the Victorian transport safety regulator as to whether the adoption of LCLCWDs would be prohibited by the Rail Safety Act 2006. The safety regulator responded with the following:

"The Rail Safety Act 2006 (RSA) requires Accredited Rail Operators (AROs) to eliminate risks and where this is not possible to reduce those risks to a standard of 'so far as is reasonably practicable'. […] If the technology being considered achieves risk to be eliminated or reduced so far as is reasonably practicable, then prima facie it could be utilized consistently with the RSA".

In 2007, an inquiry into improving safety at level crossings by the Road Safety Committee of the Parliament of Victoria determined that LCLCWDs are not, nor should be a replacement or substitute for conventional high safety integrity level crossings. The committee further recommended that these technologies be used as a supplement or enhancement to existing controls at passive crossings (e.g. stop or give-way sign). The committee noted that LCLCWDs could potentially provide better levels of risk mitigation than currently available at passive crossings. Therefore, the lack of fail-safety was not considered to be a sufficient reason in itself for their rejection.

Without legislative reform, AROs may be able to make a case for the adoption of LCLCWDs if reduction of risk so far as is reasonably practicable (SFAIRP) can be demonstrated. The National Transport Commission of Australia and Rail Safety Regulators' Panel have created a national rail safety guideline on the meaning of duty to ensure safety SFAIRP (Salter, 2008). This guideline aims to assist duty holders with interpretation of the legislative requirement and

Table 2. Level crossing stocktake (Railway Industry Safety and Standards Board, 2009)

Level crossing type	QLD	NSW	VIC	SA	WA	NT	TAS	Double Ups	Total
Pedestrian	147	239	857	243	152	7	14		1659
Public	3192	1811	1960	193	1365	61	256		8838
Private	9204	1652	n/a	90	1402	160	n/a		12508
Maintenance	241	160	n/a	1	164	n/a	n/a		566
Total	12784	3862	2817	527	3083	228	270	-39	23532
Percentage break-down	54.3%	16.4%	11.9%	2.2%	13.1%	1.0%	1.1%		100%

to provide practical guidance on how to satisfy it. Essentially SFAIRP necessitates a quantitative assessment of the costs and benefits and a judgement made as to the practicability of the proposed control measure by the margin of the disproportion.

The case study presented in this chapter makes an argument for deployment of LCLCWDs consistent with SFAIRP principles, based on treatment of a population of level crossings rather than treatment of a single crossing. The argument assumes that installation of LCLCWDs over a greater population of crossings can provide a greater safety benefit than a lesser crossing population treated with conventional high-integrity level crossing warning devices for a given investment.

The difficulty with this argument is that accidents occur at a single level crossing. On an individual crossing basis, the cost of a conventional active level crossing warning system would not be significant when compared to the capital and operational budgets of railway infrastructure owners. As a result, it would be difficult to argue from a reasonableness perspective that the conventional treatment was too costly for the business after the fact i.e. after a collision. However the railway infrastructure owner does have a budget limit with which to satisfy the obligation to reduce the collective passive crossing risk to as low as reasonably practicable. Demonstrating gross disproportion where the costs exceed the benefits should be a strong if not compelling defence.

ASSESSING RISK OF LEVEL CROSSINGS WITH PASSIVE CONTROLS

This section provides a high-level risk assessment based on averaged annualized data. The purpose of this assessment is to provide an indication of the magnitude of risk passive level crossings pose to employees, passengers and the general public; and to evaluate the viability of low cost level crossing warning devices (LCLCWDs) as a risk mitigation option.

Assessing risk is not a precise science, and in the case of rail level crossings, statistical uncertainty is a consequence of the limited number of occurrences. As Australia-wide data was not available, level crossing occurrence data from Queensland has been used as an indicator for Australia, given that more than 50% of Australia's crossings are in Queensland. Table 2 illustrates the breakdown of level crossings in Australia. The numbers include both passively and actively controlled crossings. This risk assessment focuses on reliability aspects of LCLCWDs, as reliability issues are the major obstacles impeding their adoption.

Identification of Hazards on Level Crossings with Passive Controls

A hazard is a precursor to an accident, an actual event that has occurred and has resulted in death, injury, and/or loss. The transition from hazard to

mishap is based on two factors: the set of hazards involved, and the accident risk presented by these hazards. Ericson (Ericson, 2005) defines a hazard to be comprised of three components: a hazardous element (e.g. a passive level crossing); an initiating mechanism that triggers the hazard to occur (e.g. the road vehicle and train at or near the crossing); and the target / threat, the person or object that is vulnerable to injury and/or damage (e.g. passengers, employees, general public and property).

Passive level crossings are crossings with passive warning devices, typically stop or give-way signs as a primary control and warning signs on approach to the crossing. As these types of crossings do not provide active protection, it is the responsibility of the road user to look for approaching trains. In the context of passive level crossings, the primary hazard identified was *train to motor vehicle collision at level crossing*. The hazard, causes and consequences are detailed in Table 3 and were identified using the rail infrastructure manager checklist from the Victorian safety regulator's rail operator accreditation guide (Public Transport Safety Victoria, 2006) and refined using accident mechanisms and contributing characteristics from ALCAM (Australian Level Crossing Assessment Model (ALCAM) Technical Committee, 2007). The frequency and severity of the hazards were estimated using the Hazard Ranking Matrix in Table 4. Other hazards such as derailment are not developed in this case study.

Cause-Consequence Analysis

The purpose of cause-consequence analysis is to identify and evaluate all the possible outcomes that can result from an initiating event. For hazard 4, *train to motor vehicle collision at level crossing*, the initiating event is assumed to be *road vehicle at or near level crossings when train is at or near crossing*. *Failed to stop* and *unable to avoid train* have been identified as intermediate events that can potentially lead to the hazard. For simplicity, secondary consequences such as *derailment* have been ignored.

Event probabilities have been derived from level crossing occurrence data supplied by (Department of Transport and Main Roads Queensland - Rail Safety Regulation Branch, 2010). From this data, 1268 public level crossings with passive or no control were identified. Vehicle volumes were estimated based on demographics and vehicle registrations. Train volumes were estimated from passenger and freight timetables. Table 5 details the averaged vehicle and train volume estimates for passive level crossings in Queensland.

It is estimated that an average of 7.5261 trains traverse a given passive level crossing each day (0.31359 per hour). A train is within range of the crossing when it is at least 20 seconds away; this is the specified minimum warning time for actively controlled level crossings (Standards Australia, 2007). The probability of the train being at or near a given passive level crossing is:

$$P(train\_at\_or\_near\_crossing) = (0.31359 \times 20) \div 3600 = 1.7422\times10^{-3}$$

On average, 123.3652 vehicles use a given passive level crossing each day (5.1402 per hour). Assuming the vehicle takes approximately 5 seconds to traverse the crossing, the probability of a vehicle being present at the crossing is:

$$P(vehicle\_at\_crossing) = (5.14022 \times 5) \div 3600 = 7.1392\times10^{-3}$$

The probability of a vehicle being near or at a given passive level crossing when a train is near or at the crossing is:

$$P(vehicle\_and\_train\_at\_crossing) = P(train\_at\_or\_near\_crossing) \times P(vehicle\_at\_crossing)$$

$$= 1.7422\times10^{-3} \times 7.1392\times10^{-3} = 1.2438\times10^{-5} = P(IE)$$

Table 3. Hazard identification list for a level crossing with passive controls

No.	Hazard	Causes	Est. Freq.	Est. Severity	Hazard Rank	Comments
1	Derailment	Heavy road vehicle collides with train (escalation of hazard 4); other causes not developed	…	…	…	Potential to cause other hazards e.g. toxic chemical spill, fire, explosion, etc. *Not developed in this case study*
2	Train to train collision	…	…	…	…	*Not developed in this case study*
3	Train to object collision	…	…	…	…	*Not developed in this case study*
4	Train to motor vehicle collision at level crossing	Road user does not perceive train near or at crossing. Contributing characteristics: • Visibility reduced by vegetation, buildings, track curvature, another stopped train, etc. • Possible sun glare • Crossing signage is not understood, obscured by vegetation, buildings, road curvature • Road user fatigue • Seasonal / infrequent train patterns • Train unexpected due to low frequency of trains Racing train or misjudged train speed. Contributing characteristics: • High train speed (misjudged train speed) • Low train speed (racing train) • Long train length (e.g. 2km coal train) Road user unable to stop in time. Contributing characteristics: • Insufficient stopping sight distance • Loose road surface / poor road surface conditions	2	5	10	Potential for multiple fatalities; Derailment hazard can result from collision with heavy vehicles (see Hazard 1)
5	Pedestrian struck at level crossing	…	…	…	…	*Not developed in this case study*
…	…	…	…	…	…	...

Table 4. Hazard ranking matrix

			Estimated Hazard Severity				
			Minor injury	**Major injury**	**Multiple major injuries**	**Single fatality**	**Multiple fatalities**
			1	**2**	**3**	**4**	**5**
Estimated Hazard Frequency	**Daily to monthly**	**5**	Medium 5	Medium 10	High 15	Extreme 20	Extreme 25
	Monthly to yearly	**4**	Medium 4	Medium 8	High 12	High 16	Extreme 20
	Once every year to 10 yearly	**3**	Low 3	Medium 6	Medium 9	High 12	High 15
	Once every 10 to 100 years	**2**	Low 2	Medium 4	Medium 6	Medium 8	Medium 10
	Less than once every 100 years	**1**	Low 1	Low 2	Low 3	Medium 4	Medium 5

Table 5. Vehicle and train volumes per level crossing with passive controls in Queensland

Passive level crossings traffic volumes	Average (μ)	Standard deviation (σ)
Vehicle volume (vehicles / day)	123.3652	188.9922
Train volume (trains / day)	7.5261	7.8212

This is the probability of the initiating event (IE) used for the cause-consequence analysis. The intermediate event probabilities are calculated below.

From the level crossing occurrence data provided by (Department of Transport and Main Roads Queensland - Rail Safety Regulation Branch, 2010), it has been estimated that there are on average 0.22904 near miss occurrences and 2.1370×10^{-2} collisions on all passive level crossings in Queensland per day. Note that the number of near misses reported may not be consistent with the actual number of near misses due to under reporting and the subjective nature of self-reporting near-miss occurrences.

For a single passive level crossing, there are 1.8063×10^{-4} near misses and 1.6853×10^{-5} collisions per day. The probability that a vehicle near or at a passive level crossing is involved in a near-miss incident or collision is as follows:

$$P(near\_miss) = 1.8063\times10^{-4} \div 123.3652 = 1.4642\times10^{-6}$$

$$P(collision) = 1.6853\times10^{-5} \div 123.3652 = 1.3661\times10^{-7}$$

Based on the above consequence probabilities, event probabilities can be calculated for the intermediate events *failed to stop* and *unable to avoid train*. The conditional probability that the road user fails to perceive the train and make a controlled stop, given a that the vehicle is near or at a given passive level crossing and a train is near or at the crossing, is as follows:

$$P(failed\_to\_stop|IE) = (1.4642\times10^{-6} + 1.3661\times10^{-7}) \div 1.2438\times10^{-5} = 0.128708$$

The probability that the vehicle is not able to successfully take emergency action to avoid the train is as follows:

$$P(unable\_to\_avoid\_train|failed\_to\_stop) = 1.3661\times10^{-7} \div (1.4642\times10^{-6} + 1.3661\times10^{-7}) = 8.5339\times10^{-2}$$

Figure 2 illustrates the cause-consequence model for a passive level crossing, based on the above probability calculations.

Using the cause-consequence model, the probability of an outcome (i.e. *safe condition*, *near miss* or *road user collides with train*) can be calculated as:

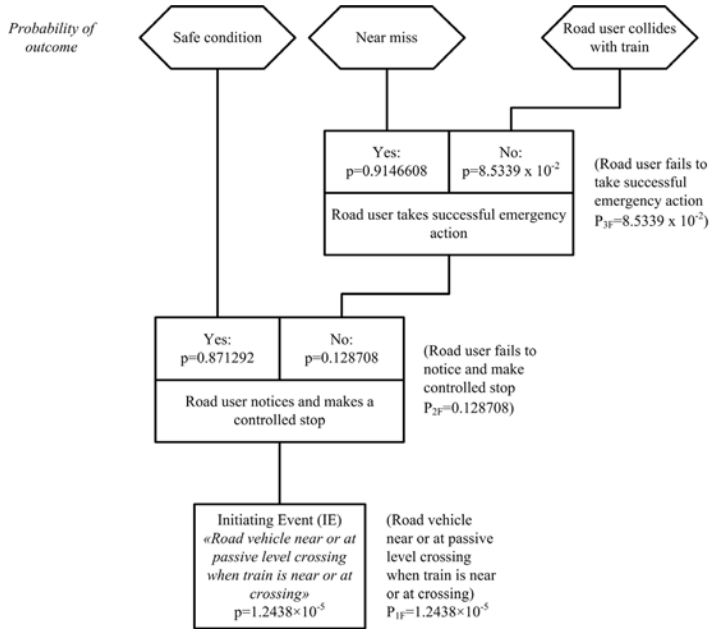
$$P_{SAFE\_CONDITION} = P_{1F} \times (1\text{-}P_{2F}) = 1.0837\times10^{-5}$$

$$P_{NEAR\_MISS} = P_{1F} \times P_{2F} \times (1\text{-}P_{3F}) = 1.4642\times10^{-6}$$

$$P_{ROAD\_USER\_COLLIDES\_WITH\_TRAIN} = P_{1F} \times P_{2F} \times P_{3F} = 1.3661\times10^{-7}$$

The cause-consequence model can be used to evaluate options that improve the road user's ability to perceive a train and make a controlled stop or the ability to take successful emergency action. The model illustrated in Figure 2 is trivial, however, was deemed sufficient for the analysis presented in this chapter. The intent of the case study is to measure improvement to the success of the *road user notices and makes a controlled stop* gate with the installation of LCLCWDs. As actual collision and fatality data is used in later analysis, a comprehensive cause-consequence model was not considered necessary. A more comprehensive model would elaborate the range of causes and consequences taking into account the types of trains and vehicles involved and their cargo, collision speeds, condition of the road surface, human factors, crossing topology and

Figure 2. Cause-consequence model for initiating event "Road vehicle near or at passive level crossing when train is near or at crossing"



environmental factors. Modelling of these factors was not necessary, besides which would require detailed analysis of level crossing incidents, requiring access to incident reports. Such reports were not available.

Analysis of Potential Safety Losses

Loss analysis is used to determine the magnitude of potential safety losses associated with each hazard. The consequences (outcomes) identified in the cause-consequence analysis are used to determine the potential safety losses. It is assumed that the *safe condition* outcome does not result in any losses. While small losses can occur from a near-miss incident, this outcome does not result in any safety loss in terms of person equivalent fatalities. Table 6 details the level crossing occurrences that occurred on public level crossings with passive or no control in Queensland over the last 5 years.

Using the level crossing occurrence data above, the person equivalent fatalities (PEF) per collision can be calculated. This value corresponds to injuries and fatalities associated with each collision. It is assumed for this analysis that that 1 fatality is equivalent to 10 major injuries. Table 7 details the PEFs calculated for each group of interest.

Based on the above values, the PEF for a single collision in Queensland is therefore 0.12893 fatalities per collision. No passenger injuries or fatalities have been recorded between 2006 and 2010. This is probably due to the small number of passenger services that run on lines with passive level crossings. Table 8 and Table 9 detail the safety losses for the *collision* outcome for employees and the general public.

The annual frequency has been determined by averaging the level crossing occurrence values in Table 6. These values can also be calculated by multiplying the volume of vehicles for all passive level crossings in Queensland with the probabil-

Table 6. Level crossing occurrences on public level crossings with passive controls (Department of Transport and Main Roads Queensland - Rail Safety Regulation Branch, 2010)

	2010	2009	2008*	2007^	2006^
Passenger Fatalities	0	0	0	0	0
Employee Fatalities	0	0	0	0	0
General public Fatalities	1	1	2	0	0
Passenger Serious Injuries	0	0	0	0	0
Employee Serious Injuries	0	0	1	0	0
General public Serious Injuries	1	3	1	0	1
Collisions	7	6	8	6	12
Near Misses	79	91	39	-	-

^Note that in 2006 and 2007, near-miss occurrences were not collected by the Department of Transport and Main Roads Queensland.
*Note that in 2008 near-miss occurrences were only collected from July onwards.

ity that the vehicle is involved in a near-miss or collision:

$$F(collision) = 123.2652 \times 1268 \times 365 \times P(collision) = 7.8 \text{ per annum}$$

$$F(near\_miss) = 123.2652 \times 1268 \times 365 \times P(near\_miss) = 83.6 \text{ per annum}$$

This calculation method is important for calculating the mitigated safety losses for the options analysis. In this case, the probabilities of collision and near miss are obtained from the cause-consequence analysis, which may be updated with various mitigation options.

Analysis of Options

This analysis identifies potential risk mitigation options for each hazard. The most common and obvious improvement that can be made at passive level crossings is the provision of active control. This effectively eliminates the need for the road user to make a decision as to whether it is safe to traverse the crossing. Elvik et al (Elvik, Høye, Vaa, & Sørensen, 2009) estimate that the addition of flashing lights and sound signals at level crossings that previously had passive warning signs was estimated to reduce the number of collisions by 51%, based on a number of international before and after studies. The addition of boom barriers was estimated

Table 7. Person equivalent fatalities (PEF) per collision

Fatalities and serious injuries / collision	2010	2009	2008	2007	2006	Average
Employee Fatalities	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
Employee Serious injuries	0.00000	0.00000	0.12500	0.00000	0.00000	0.02500
Employee PEF	**0.00000**	**0.00000**	**0.01250**	**0.00000**	**0.00000**	**0.00250**
General Public Fatalities	0.14286	0.16667	0.25000	0.00000	0.00000	0.11190
General Public Serious injuries	0.14286	0.50000	0.12500	0.00000	0.08333	0.17024
General Public PEF	**0.15714**	**0.21667**	**0.26250**	**0.00000**	**0.00833**	**0.12893**

Table 8. Safety losses per incident and per annum for all passive level crossings in Queensland

Incident	Frequency / annum	Safety loss (PEF) per incident		Safety loss per annum (PEF)	
		Employee	Public	Employee	Public
Collision	7.8	0.00250	0.12893	0.01950	1.00564
Near-miss	83.6	-	-	-	-

Table 9. Safety losses per incident and per annum for a single passive level crossing

Incident	Frequency / annum	Safety loss (PEF) per incident		Safety loss per annum (PEF)	
		Employee	Public	Employee	Public
Collision	6.1514×10^{-3}	0.00250	0.12893	1.5379×10^{-5}	7.9309×10^{-4}
Near-miss	6.5931×10^{-2}	-	-	-	-

to further reduce the number of collisions by 45%. Table 10 details the risk mitigation options considered in this analysis. Cost estimates of these options are based on a number of low-cost product offerings. The difference between options 1a, b and c is the level of safety integrity provided. All options provide the same interface to the road user.

It is estimated that the installation of the above options containing RX5 flashing light assemblies with no boom gates reduce collisions by 51%. This level of reduction is based on the performance of high-integrity conventional level crossing warning devices. As LCLCWDs are expected to be inherently less safe, the estimated safety benefit must be reduced to take this into account. The following assumptions have been made in calculating the safety benefit in terms of mitigated safety loss for each of the options:

- If a warning device fails right-side (safe failure state or train detected when no train is approaching), there is a safety benefit in that flashing lights are likely to cause a road user to stop and proceed with caution;
- If a warning device fails right-side frequently or for prolonged periods of time, there may be an increase in the likelihood of road users engaging in risky driving behaviour at subsequent encounters with

Table 10. Risk mitigation options

Hazard ref.	Hazard	Option	Estimated Cost
1	Road user fails to detect crossing and give way to train when train is approaching level crossing	1a. LCLCWD with RX5 flashing light assemblies (very low safety integrity)	Initial cost of $75,000 + $5,400 maintenance per annum
		1b. LCLCWD with RX5 flashing light assemblies (low safety integrity)	Initial cost of $100,000 + $5,400 maintenance per annum
		1c. LCLCWD with RX5 flashing light assemblies (medium safety integrity)	Initial cost of $150,000 + $5,400 maintenance per annum
		2. Conventional LCWD with RX5 flashing light assemblies (high safety integrity)	Initial cost of $400,000 + $10,000 maintenance per annum

the same crossing and potentially other crossings. This effect is excluded from this analysis;

- If a warning device fails wrong-side, there is no safety benefit to the road user. A wrong-side failure has the potential to significantly increase the risk of a near-miss or collision, as road users are likely to assume that the lack of warning indicates that there is no train approaching. Wrong-side failure of the a level crossing warning device in this analysis refers to the event *warning not provided when train is approaching or warning not provided in time*; and
- The RX5 flashing light assemblies of LCLCWDs will be the same as those defined in AS1742.7-2007 (Standards Australia, 2007). This eliminates the need to estimate the change in risk from a road-user human factors perspective and constrains the comparison to the underlying technology.

In order to determine the mitigated safety loss facilitated by the installation of a LCLCWD at a level crossing with passive controls, the wrong-side failure probability and frequency are calculated based on an average crossing exposure. As a wrong-side failure only occurs when a demand is made, it is reasonable to consider failure probabilities on demand.

For autonomous high-integrity level crossing warning devices not associated with railway signalling interlocking i.e. one reliant totally on detection of trains using track circuits, an assumption has been made that such systems will not have a higher safety integrity than the integrity (reliability) of the track circuits for detecting the train. Based on experience and historical reliability data of track circuits, the rate of failure on demand of a track circuit was estimated to be 5.0×10^{-7}. While high-integrity level crossing warning devices may have SIL4 elements and be failsafe, it is incorrect to assume that this inherent high integrity performance applies to the detection of a train. There have been incidents where a track circuit has failed to detect a train. For the purpose of this analysis, a track circuit is considered to be no better than SIL2.

Calculations have been made based on rates of failure on demand for the high-integrity level crossing warning device, and for three low-cost options 1a, b and c; the low-cost options have a lower inherent safety integrity by a factor of 1000, 100 and 10 respectively compared to the conventional technology. The calculations for option 1b are illustrated below, and the results for the three options are detailed in Table 11.

The estimated frequency of wrong-side failures per annum for option 1b is detailed below. The calculation uses 5.0×10^{-5} as the probability of failure on demand, which is option 1b.

F(wrong_side_failure) = PFD × average demands/annum

$5.0\times10^{-5} \times 2747.0265 = 1.3735\times10^{-1}$ per annum

The frequency at which a train and vehicle are near or at the crossing in wrong-side failure is:

F(vehicle_at_crossing_in_wrong_side_failure)

= P(vehicle_at_crossing) × F(wrong_side_failure)

$= 7.1392\times10^{-3} \times 1.3735\times10^{-1} = 9.8058\times10^{-4}$ per annum

The probability of failure per hour (PFH) can be calculated from the PFD given the number of demands per hour:

PFH = PFD × (average_demands_per_annum ÷ 8760)

$= 5.0\times10^{-5} \times 0.3135875 = 1.5679\times10^{-5}$

Table 11. Probability of wrong-side failure on demand and per hour for the mitigation options

	Option 1a	Option 1b	Option 1c	Option 2
PFD (probability of dangerous ***failure on demand***) of level crossing (i.e. warning not provided or provided too late)	5.0×10^{-4}	5.0×10^{-5}	5.0×10^{-6}	5.0×10^{-7}
Average demands per crossing per annum	2747.0265	2747.0265	2747.0265	2747.0265
Estimated frequency of wrong-side failure per annum	1.3735	1.3735×10^{-1}	1.3735×10^{-2}	1.3735×10^{-3}
Frequency of vehicle and train at crossing in wrong-side failure	9.8058×10^{-3}	9.8058×10^{-4}	9.8058×10^{-5}	9.8058×10^{-6}
PFH (probability of dangerous ***failure per hour***) of level crossing (i.e. warning not provided or provided too late)	1.5679×10^{-4}	1.5679×10^{-5}	1.5679×10^{-6}	1.5679×10^{-7}

The mitigated safety loss for a level crossing warning device is determined by calculating the reduction of person equivalent fatalities (PEF) after installation. This calculation is based on the reduction estimate of 51% for the installation of flashing lights at a level crossing with passive controls and the reduction of safety benefit due to wrong-side failures.

To estimate the reduction of safety benefit due to wrong-side failure, it is assumed that a collision will occur when a vehicle and train are at the crossing in wrong-side failure. The additional PEF per annum due to wrong-side failure of the warning device is calculated as follows (example for Option 1a):

PEF(<group>) × F(vehicle_at_crossing_in_wrong_side_failure)

PEF(wrong_side_failure_employees) = $0.00250 \times 9.8058\times10^{-3} = 2.4514\times10^{-5}$

PEF(wrong_side_failure_public) = $0.12893 \times 9.8058\times10^{-3} = 1.2642\times10^{-3}$

Estimates of the mitigated losses per annum in terms of PEF are detailed in Tables 12, 13, 14, and 15.

From Table 9, the unmitigated per annum collision frequency is 6.1514×10^{-3}, which reduces to 3.0173×10^{-3} with Option 1a. Installation of option 1a reduces the number of collisions by 50.95% ($0.51 - 5\times10^{-4}$) taking into account the reduction for wrong-side failures, but not taking into account the additional risk of collision due to wrong-side failures. This additional risk is subsequently added using the method described above. Note that the negative mitigation values for Option 1a demonstrate that the risk will be increased. This is because the road user now relies on the low-cost active control technology (i.e. if it does not warn of an approaching train, the road user will assume that there is no train approaching). For this reason, option 1a is not considered in Table 13.

Installation of option 1b reduces the number of collisions by 50.995% ($0.51 - 5\times10^{-5}$) taking into account the reduction for wrong-side failures, but not taking into account the additional risk of collision due to wrong-side failures.

Installation of option 1c reduces the number of collisions by 50.9995% ($0.51 - 5\times10^{-6}$) taking into account the reduction for wrong-side failures, but not taking into account the additional risk of collision due to wrong-side failures.

Installation of option 2 reduces the number of collisions by 50.99995% ($0.51 - 5\times10^{-7}$) taking into account the reduction for wrong-side failures, but not taking into account the additional risk of collision due to wrong-side failures.

Table 12. Option 1a Low-Cost LCWD with RX5 flashing light assembly (probability of wrong-side failure on demand of 5×10^{-4})

Incident	Frequency / annum	Safety loss per incident (PEF)		Safety loss per annum (PEF)	
		Employee	Public	Employee	Public
Collision	3.0173×10^{-3}	2.5000×10^{-3}	1.2893×10^{-1}	7.5432×10^{-6}	3.8901×10^{-4}
Near-miss	-	-	-	-	-
Losses per annum due to wrong-side failure				2.4514×10^{-5}	1.2642×10^{-3}
Total losses per annum with mitigation (A)				3.2058×10^{-5}	1.6533×10^{-3}
Total losses per annum without mitigation (B)				1.5379×10^{-5}	7.9309×10^{-4}
Total mitigated losses per annum (B-A)				**-1.6679×10^{-5}**	**-8.6016×10^{-4}**

Table 13. Option 1b: Low-Cost LCWD with RX5 flashing light assembly (probability of wrong-side failure on demand of 5×10^{-5})

Incident	Frequency / annum	Safety loss per incident (PEF)		Safety loss per annum (PEF)	
		Employee	**Public**	**Employee**	**Public**
Collision	3.0145×10^{-3}	2.5000×10^{-3}	1.2893×10^{-1}	7.5363×10^{-6}	3.8866×10^{-4}
Near-miss	-	-	-	-	-
Losses per annum due to wrong-side failure				2.4514×10^{-6}	1.2642×10^{-4}
Total losses per annum with mitigation (A)				9.9877×10^{-6}	5.1508×10^{-4}
Total losses per annum without mitigation (B)				1.5379×10^{-5}	7.9309×10^{-4}
Total mitigated losses per annum (B-A)				**5.3908×10^{-6}**	**2.7801×10^{-4}**

Table 14. Option 1c: Low-Cost LCWD with RX5 flashing light assembly (probability of wrong-side failure on demand of 5×10^{-6})

Incident	Frequency / annum	Safety loss per incident (PEF)		Safety loss per annum (PEF)	
		Employee	**Public**	**Employee**	**Public**
Collision	3.0142×10^{-3}	2.5000×10^{-3}	1.2893×10^{-1}	7.5356×10^{-6}	3.8862×10^{-4}
Near-miss	-	-	-	-	-
Losses per annum due to wrong-side failure				2.4514×10^{-7}	1.2642×10^{-5}
Total losses per annum with mitigation (A)				7.7807×10^{-6}	4.0126×10^{-4}
Total losses per annum without mitigation (B)				1.5379×10^{-5}	7.9309×10^{-4}
Total mitigated losses per annum (B-A)				**7.5978×10^{-6}**	**3.9183×10^{-4}**

Table 15. Option 2. Conventional LCWD with RX5 flashing light assembly (probability of wrong-side failure on demand of 5×10^{-7})

Incident	Frequency / annum	Safety loss per incident (PEF)		Safety loss per annum (PEF)	
		Employee	Public	Employee	Public
Collision	3.0142×10^{-3}	2.5000×10^{-3}	1.2893×10^{-1}	7.5355×10^{-6}	3.8862×10^{-4}
Near-miss	-	-	-	-	-
Losses per annum due to wrong-side failure				2.4514×10^{-8}	1.2642×10^{-6}
Total losses per annum with mitigation (A)				7.5600×10^{-6}	3.8988×10^{-4}
Total losses per annum without mitigation (B)				1.5379×10^{-5}	7.9309×10^{-4}
Total mitigated losses per annum (B-A)				**7.8185×10^{-6}**	**4.0321×10^{-4}**

In order to determine the monetary value of mitigated safety losses, human costs, property damage and other related costs need to be estimated. The monetary value for preventing a fatality (VPF) of $6,287,873 was used consistent with the estimate of human costs based on the willingness to pay method used in the railway level crossing incident costing model for the Australian railway industry safety and standards board (RISSB) (Tooth & Balmford, 2010).

Estimates for property damage and other costs associated with level crossing accidents were obtained from (Tooth & Balmford, 2010) and are detailed in Table 16.

Table 17 details the mitigated costs based on the estimated costs per crossing accident and the mitigated collisions due to installation of the level crossing warning device. Table 18 details the monetary value of total mitigates losses per crossing.

Table 19 details costs and benefits of a population treatment of options 1b and 2 over a period of 25 years with an annual capital investment of $10,000,000. Note that figures represent net present value (NPV) of costs and benefits. The interest rate used for escalation of figures was 3.066%, an average of the consumer price index (CPI) from 2008 to 2010. An Australian track owner and infrastructure manager provided esti-

Table 16. Estimated costs per crossing accident (Tooth & Balmford, 2010)

Description	Incidence	Total estimated cost in 2010	Estimated average cost per incident
Property damage			
Rail fixed structures and rolling stock	70	$4,240,198.00	$60,574.26
Road vehicle	60	$1,524,502.00	$25,408.37
Other costs			
Police and emergency services	70	$557,341.00	$7,962.01
Delay costs	70	$593,519.00	$8,478.84
Safety investigation	70	$236,760.00	$3,382.29
Insurance administration and legal	70	$3,156,806.00	$45,097.23
Road vehicle unavailability	60	$93,858.00	$1,564.30
Totals		**$10,402,984.00**	**$152,146.30**

Table 17. Mitigated costs due to collisions per crossing

	Option 1b	Option 1c	Option 2
Frequency of collisions per crossing without mitigation per annum (N)	6.1514×10^{-3}	6.1514×10^{-3}	6.1514×10^{-3}
Cost of collisions per crossing per annum (A=N×$152,146.30)	**$937.89**	**$937.89**	**$937.89**
Estimated frequency of collisions per crossing with mitigation per annum (I)	3.0145×10^{-3}	3.0142×10^{-3}	3.0142×10^{-3}
Estimated frequency of collisions per crossing due to wrong-side failure (J)	9.8058×10^{-4}	9.8058×10^{-5}	9.8058×10^{-6}
Estimated total frequency of collisions per crossing with mitigation per annum (K=I+J)	3.9951×10^{-3}	3.1123×10^{-3}	3.0240×10^{-3}
Cost of collisions with mitigation per crossing per annum (B=K×$152,146.30)	**$609.12**	**$474.52**	**$461.06**
Mitigated costs due to collisions (A-B)	**$328.77**	**$463.37**	**$476.83**

mates of maintenance costs. The population treatment aims to install level crossing warning devices on 75% of passive level crossings in Queensland, approximately 950 out of a total of 1268 level crossings with passive controls. The 75% figure represents the number of crossings with an exposure score less-than or equal to that calculated using the average train and vehicle volumes. The exposure score is calculated as follows:

$$\text{exposure\_score} = \text{train\_volume\_per\_day} \times \text{vehicle\_volume\_per\_day}$$

$$\textit{exposure\_score\_avg} = 123.3652 \times 7.5261 = 928.4588$$

A cost-benefit ratio (CBR) greater than 1 indicates that the benefit is greater than the cost. A CBR less than 1 indicates that the cost is greater than the benefit, where values less than 0.1 indicate gross disproportion to the benefit. Figure 3 illustrates the benefits versus costs for Options 1b, 1c and 2. Figure 4 compares the cumulative benefits and cumulative costs for Options 1b, 1c and 2.

Factors not taken into account in the cost benefit analysis include cost of lost opportunity and costs of delays on the rail operator due to right-side failures. A more thorough cost-benefit analysis will be possible after the lifecycle assessment criteria has been developed by the Rail CRC Affordable Level Crossings project. These crite-

Table 18. Monetary value of total mitigated losses per crossing

	Option 1b	Option 1c	Option 2
Mitigated costs due to collisions (A)	**$328.77**	**$463.37**	**$476.83**
Mitigated safety losses (PEF) employee (I)	5.3908×10^{-6}	7.5978×10^{-6}	7.8185×10^{-6}
Mitigated safety losses (PEF) general public (J)	2.7801×10^{-4}	3.9183×10^{-4}	4.0321×10^{-4}
Total mitigated safety losses (PEF) (K=I+J)	2.8340×10^{-4}	3.9943×10^{-4}	4.1103×10^{-4}
Monetary value of mitigated safety loss (VPF) (B=K×$6,287,873.00)	$1,782.01	$2,511.56	$2,584.52
Mitigated total loss per crossing (A+B)	**$2,110.78**	**$2,974.93**	**$3,061.34**

Table 19. Costs and benefits of population treatment of level crossings with passive control over 25 years

	Mitigation Option 1b	Mitigation Option 1c	Mitigation Option 2
Number of crossings installed over 25 year period (out of 950)	950	950	625
Number installed per year	100	66	25
Installation cost (Year 1 base)	$100,000.00	$150,000.00	$400,000.00
Annual maintenance cost (Year 1 base)	$5,400.00	$5,400.00	$10,000.00
NPV of benefit over 25 years	$40,579,807.76	$50,291,176.18	$23,916,747.42
NPV of total expenditure over 25 years	$198,815,000.00	$233,787,000.00	$328,125,000.00
NPV of net benefit over 25 years	$-158,235,192.24	$-183,495,823.82	$-304,208,252.58
BCR (Benefit Cost Ratio)	**0.2041**	**0.2151**	**0.07289**

Figure 3. Costs and benefits for LCLCWDs vs. conventional LCWDs



Figure 4. Comparison of cumulative costs and benefits for LCLCWDs vs. conventional LCWDs

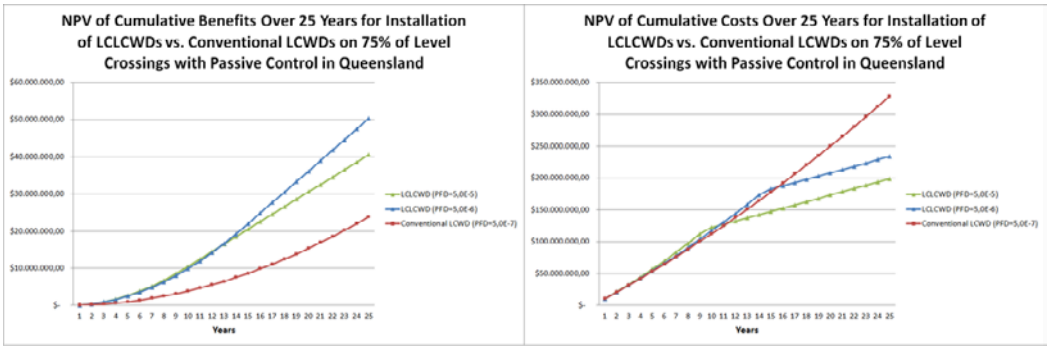
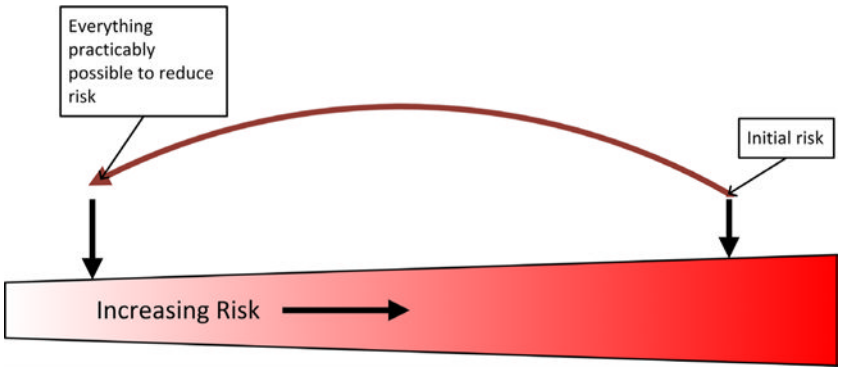


Figure 5. The SFAIRP Requirement



ria will allow cost over the lifetime of a LCLCWD to be determined more accurately.

Demonstrating SFAIRP Applicability

In demonstrating that risks have been reduced So Far As Is Reasonably Practicable (SFAIRP), as required by rail safety legislation, the following factors must be considered (Salter, 2008):

- The likelihood of the risk concerned eventuating;
- The degree of harm that would result if the risk eventuated;
- What was known or ought reasonably to be known about the risk and any ways of eliminating the risk;
- The availability and suitability of ways to eliminate or reduce the risk; and
- The cost of eliminating or reducing the risk.

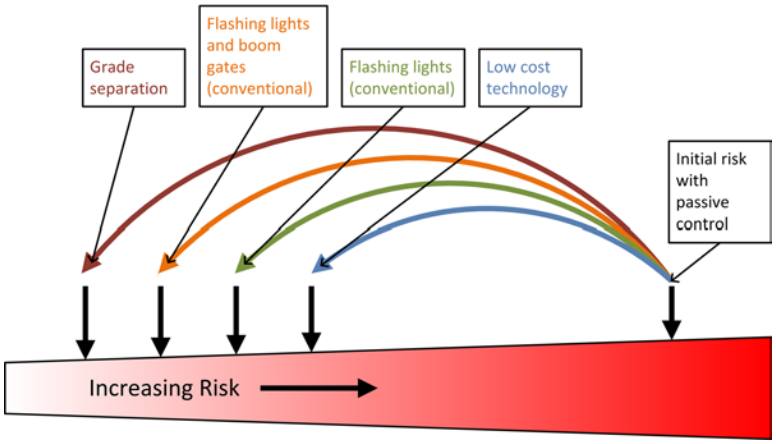
In considering the cost of eliminating or reducing the risk, practitioners must demonstrate that the likelihood of the risk eventuating is remote or that the cost is grossly disproportionate to the safety benefit. The meaning of gross disproportion was established in the UK by Lord Asquith in the case of Edwards versus The National Coal Board (1949).

"A computation must be made in which the quantum of risk is placed on one scale and the sacrifice, whether in money, time or trouble, involved in the measures necessary to avert the risk is placed in the other, and that, if it be shown that there is a gross disproportion between them, the risk being insignificant in relation to the sacrifice, the person upon whom the duty is laid discharges the burden of proving that compliance was not reasonably practicable."

A number of options are available for reducing risk at level crossings with passive control. In this case study, it is assumed that the cost of grade separation for low exposure level crossings with passive controls would be grossly disproportionate to the safety benefit (see Figure 5). In evaluating options involving installation of active warning devices, it is difficult to differentiate between conventional and low-cost treatments using the gross disproportion test on a per crossing basis, because the risk at an individual passive crossing is small and therefore the safety benefit to be gained is also small. Figure 6 illustrates the risk reduction from a single level crossing treatment perspective.

By considering the treatment of a population of level crossings with passive controls, the installation of conventional high-integrity warning devices at level crossings with passive controls has been demonstrated to have a benefit cost ratio

Figure 6. SFAIRP Applied to a Single Level Crossing



of 0.07, indicating that the cost of installing conventional active protection is more than 10 times the safety benefit, and can therefore be considered as grossly disproportionate. For a given budget, the use of low-cost technology provides a higher net safety benefit for a population treatment than the equivalent conventional treatment for the same budget. Figure 7 illustrates the risk reduction from a population treatment perspective.

As accidents and fatalities occur at single level crossings, the argument for risk reduction for a population treatment versus treatment of a single level crossing has to be accepted by the safety regulator and law courts.

Meeting Safety Targets

Based on the figures presented in this case study, safety targets have been chosen for varying levels of exposure. Table 20 details the levels of crossing exposure considered, and the estimated percentage of level crossings with passive controls in Queensland represented by each level.

For each exposure level and range of probability of failure on demand (PFDs), calculations were performed to validate that there is a positive safety benefit and that the gross disproportion still holds (Table 21). Note that the figures in brackets refer to the range of benefit cost ratio (BCR) cor-

Figure 7. SFAIRP Applied to Population of Level Crossings

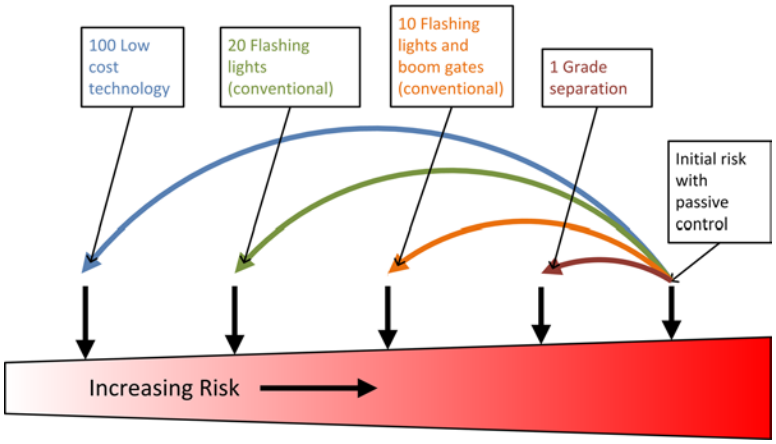


Table 20. Level crossing exposure

Level crossing exposure	Exposure score *(vehicle_volume_per_day × train_volume_per_day)*	Estimated % of passive level crossings in Queensland
Level 1	≤ 500	65%
Level 2	≤ 1000	75%
Level 3	≤ 2000	85%
Level 4	≤ 3000	95%
Level 5	≤ 6000	100%

Table 21. LCLCWD safety targets

Level crossing exposure	Safety target (PFD) for event "warning not provided when train approaching or not provided in time"			
	$\geq 1\times10^{-4}$ to $< 1\times10^{-3}$ **(Option 1a)**	$\geq 1\times10^{-5}$ to $< 1\times10^{-4}$ **(Option 1b)**	$\geq 1\times10^{-6}$ to $< 1\times10^{-5}$ **(Option 1c)**	$\geq 1\times10^{-7}$ to $< 1\times10^{-6}$ **(Option 2)**
Level 1	No	Yes* (0.198 to 0.274)	Yes (0.215 to 0.220)	Yes (0.073 to 0.073)
Level 2	No	Yes* (0.114 to 0.267)	Yes (0.208 to 0.220)	Yes (0.073 to 0.073)
Level 3	No	No	Yes (0.196 to 0.218)	Yes (0.072 to 0.073)
Level 4	No	No	Yes (0.183 to 0.217)	Yes (0.072 to 0.073)
Level 5	No	No	Yes (0.145 to 0.213)	Yes (0.071 to 0.073)

Table 22. Example top-level Safety Requirement for LCLCWDs in Queensland

Maximum tolerable PFD for event "warning not provided when train approaching or not provided in time"	**$\leq 1\times10^{-5}$**
Level crossing exposure ***(train_volume_per day × vehicle_volume_per_day)***	≤ 1000
Number of tracks	Single track only

responding to the PFD range. Figure 8 provides a comparison of mitigated safety losses by PFD and crossing exposure. The shaded region in the graphs refers to a negative safety benefit. Treatment options that appear in this region will make the crossing risk worse than if left untreated due to wrong-side failures misleading road users.

Note that in order for option 1b to be accepted, the cost of both option 1c and option 2 would have to be demonstrated to be grossly disproportionate to the safety benefit they provide. This may be the case if it is determined that option 1c cannot meet the safety target for the estimated cost of $100,000 per crossing.

From the above safety targets, high-level safety requirements for LCLCWDs in Queensland can be specified. Table 22 illustrates an example of a high level requirement for the top-level event, *warning not provided when train approaching or not provided in time*, with associated assumptions relating to level crossing exposure.

In order to demonstrate a given low-cost technology meets the above requirements, technology providers would be required to present a safety case providing evidence that the LCLCWD meets the safety requirements in the target environment and context. The safety case brings together many

Figure 8. Comparison of mitigated safety losses by PFD and level crossing exposure

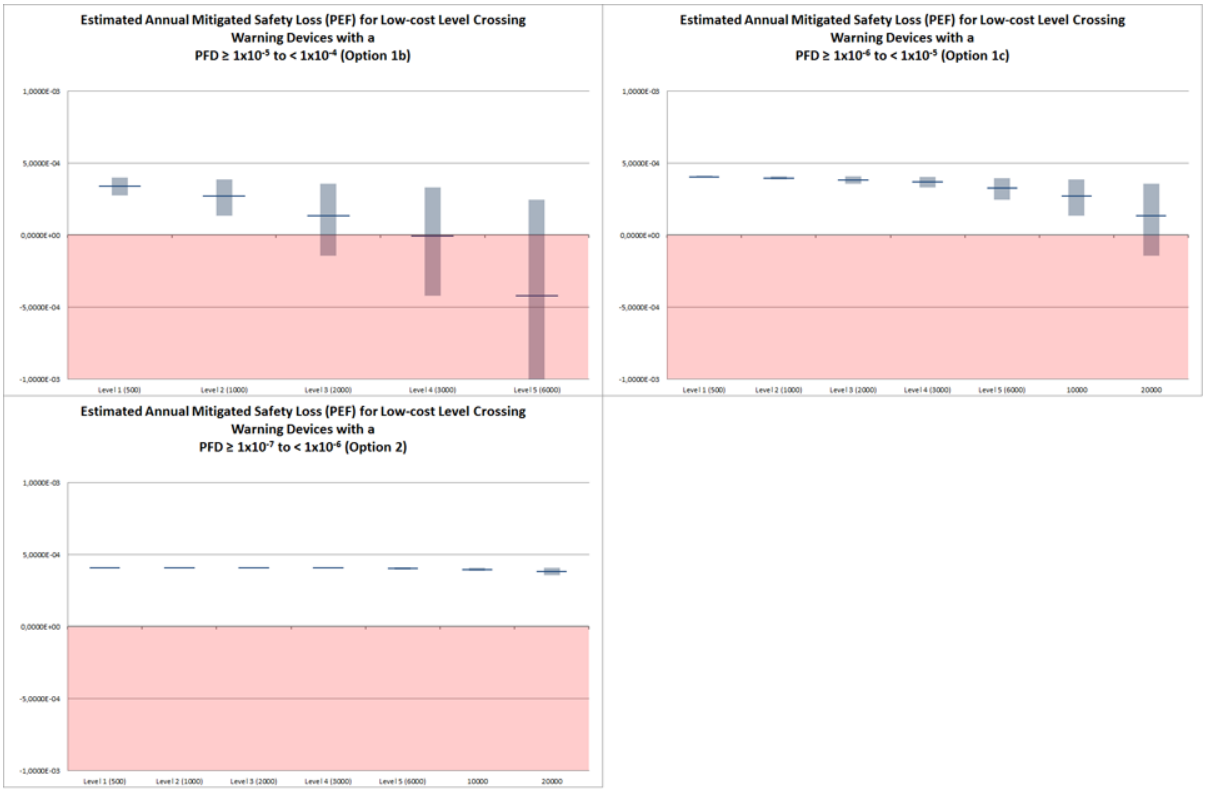
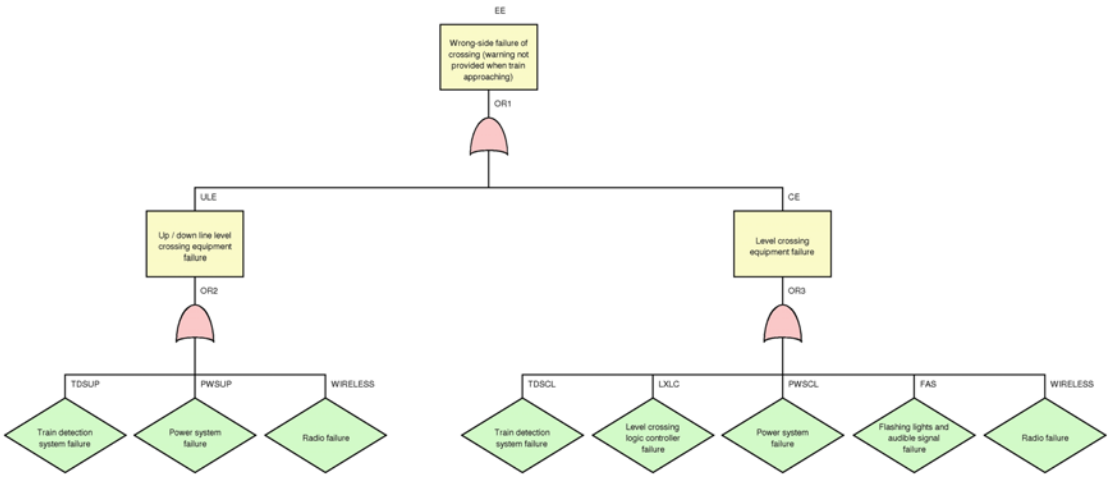


Figure 9. Example high-level fault tree



forms of evidence, that together, present a coherent argument of safety. Such evidence includes process evidence such as conformance to safety requirements, development to a target safety integrity level (SIL) for software systems, etc. and direct evidence through homologation testing, demonstrating that the device functions under the full set of operational conditions, degraded modes, failure conditions and transitions between conditions.

Developing a safety case for new technologies can be difficult, as there is often little operational experience, if any, and failure modes of these technologies are not easily understood, in part due to the increasing use of software (e.g. implementation of algorithms used in Doppler radar for train detection). Not only do novel technologies introduce new hazards, they often expose vulnerabilities that were not present with conventional technologies (e.g. wireless communication technologies used to replace fixed wiring are vulnerable to interference, whether by unintentional spurious transmissions or intentional RF jamming attacks).

Techniques such as fault tree analysis (FTA) can be used to demonstrate the LCLCWD meets the top-level safety target (PFD). Where software is involved, e.g. for programmable logic controllers (PLC) used in LCLCWDs, development to SIL levels associated with a given level of random failure aims to minimize systematic failures and provide evidence of rigour in the design and development processes (see Figure 9).

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This case study has made a basic argument for the adoption of LCLCWDs in Australia using a risk-based approach, based on a quantitative assessment of costs and benefits for the treatment of a population of level crossings with passive controls. There are a number of aspects of the case study that need to be investigated further in order to support the argument. Such aspects include analysing the effects of frequent or prolonged unavailability of LCLCWDs on road user behaviour, with the aim of determining appropriate availability targets.

A detailed estimation of lifecycle costs is also required in order to improve the accuracy of cost projections. As many of the alternative train detection technologies used in LCLCWDs have not been tested in the context of the railway or under operating environmental conditions found at level crossings, homologation testing in addition to reliability analysis is needed to give confidence that the technology is capable of performing with a level of availability and reliability consistent with availability and safety targets.

ACKNOWLEDGMENT

The CRC for Rail Innovation (established and supported under the Australian Government's Cooperative Research Centres program) has supported this work and is supporting research to further the case for the adoption of LCLCWDs in Australia. Project No. R3.122.

REFERENCES

Australian Level Crossing Assessment Model (ALCAM) Technical Committee. (2007). *Australian Level Crossing Assessment Model Technical Manual*. ALCAM.

CENELEC - European Committee for Electrotechnical Standardization. (1999). 50126-1:1999 *Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process.* CENELEC.

Department of Transport and Main Roads Queensland - Rail Safety Regulation Branch. (2010). *Rail level crossing occurrence statistics*. Brisbane.

Department of Transport NSW. (2010). *The Australian Level Crossing Assessment Model: ALCAM in Detail.*

Elvik, R., Høye, A., Vaa, T., & Sørensen, M. (2009). *The Handbook of Road Safety Measures* (2nd ed.). Bingley, UK: Emerald Group Publishing Limited.

Ericson, C. A. (2005). *Hazard Analysis Techniques for System Safety*. Hoboken, NJ: John Wiley & Sons, Inc. doi:10.1002/0471739421

Gil, M., Multer, J., & Yeh, M. (2007). *Effects of Active Warning Reliability on Motorist Compliance at Highway-Railroad Grade Crossings*.

Jordan, P. (2006, 17 February 2006). *A Trial of a Low Cost Level Crossing Warning Device.* Paper presented at the IRSE Annual Meeting.

Public Transport Safety Victoria. (2006). *Appendix G: Risk management and 'so far as is reasonably practicable' (SFARP)*. Accreditation Guideline.

Railway Industry Safety and Standards Board. (2009). *Level Crossing Stocktake*. Retrieved from http://www.rissb.com.au/userfiles/file/RLX/2009 Level Crossing Stocktake Report.pdf

Roop, S. S., Roco, C. E., Olson, L. E., & Zimmer, R. A. (2005). *An Analysis of Low-Cost Active Warning Devices for Highway-Rail Grade Crossings* (Institute, T. T., Trans.). Texas: Texas A&M University.

Salter, P. (2008). National Guideline for the Meaning of Duty to Ensure Safety So Far As Is Reasonably Practicable *National Railway Safety Guideline*: National Transportation Commission. Retrieved from http://www.rsrp.asn.au/files/legislation/42_7.pdf

Standards Australia. (1999). *AS 61508.1-1999 Functional safety of electrical/electronic/programmable electronic safety-related systems: Part 1: General requirements*. Homebush, NSW, Australia: Standards Australia.

Standards Australia. (2007). *Manual of uniform traffic control devices Part 7: Railway crossings.*

Tooth, R., & Balmford, M. (2010). *Railway Level Crossing Incident Costing Model: Railway Industry Safety and Standards Board*. RISSB.

KEY TERMS AND DEFINITIONS

Availability: The ability of a product to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval assuming that the required external resources are provided (CENELEC - European Committee for Electrotechnical Standardization, 1999).

Level Crossing with Passive Controls: The location where a railway line and road intersect at grade, where the road interface is controlled by static warning signs (stop or give-way) that are visible by road users on approach.

Level Crossing with Active Warning: The location where a railway line and road intersect at grade, where the road interface is controlled by an automatic warning system consisting of static signage, flashing lights, and in some cases audible alarms and boom barriers. The warning system is activated by an approaching train and is deactivated once the rear of the train traverses the road.

Low Cost Level Crossing Warning Device (LCLCWD): LCLCWDs are characterized by the use of alternative technologies for train detection and connectivity; and often rely on the use of solar power. They generally are not considered to have the same fail-safe and safety integrity characteristics as the conventional technologies.

Right-Side Failure: A failure condition that results in the system entering a safer state than had the failure not been present. In the case of level crossings, this means activating the crossing controls as through a train was approaching when a failure is detected.

Safety Integrity: The likelihood of a system satisfactorily performing the required safety func-

tions under all the stated conditions within a stated period of time (CENELEC - European Committee for Electrotechnical Standardization, 1999).

Safety Integrity Level (SIL): SILs are discrete levels that have an assigned dangerous failure (i.e. wrong-side failure) probability. For systematic failures, these levels are associated with a group of methods and tools, which if applied, should provide the evidence and confidence that system meets the SIL target.

So Far As Is Reasonably Practicable (SFAIRP): This is a principle established in UK law and confirmed in Australian law and is the basis for rail safety statute law in Australia. Essentially SFAIRP necessitates a quantitative assessment of the costs and benefits and a judgement made as to the practicability of the proposed control measure by the margin of the disproportion of the costs to the benefits, the costs being more than the benefits.

Wrong Side Failure: A failure condition that results in a state that is less safe than had the failure not existed. In the case of level crossings, a failure that prevents the issue of a timely warning of an approaching train to the road user e.g. failure to detect approaching train or a failure which prevents the road warning lights from flashing.

Compilation of References

(2004). Frontmatter. InNyce, D. S. (Ed.), *Linear Position Sensors: Theory and Application*. Hoboken, NJ: John Wiley & Sons.

A. Facchinetti, F. Fossati, F. Resta, and A. Collina, (2004) Hardware in the loop test-rig for identification and control application on high speed pantographs, *Shock Vib11*(3/4),445–456.

Abdulla, P. A., Deneux, J., Stalmarck, G., Agren, H., & Akerlund, O. (2004). Designing safe, reliable systems using SCADE. *LNCS 4313: IsoLA 2004,* (pp. 115–129), Paphos, Cyprus. Berlin, Germany: Springer.

Abdulla, P., Deneux, J., Stålmarck, G., Ågren, H., & Åkerlund, O. (2006). Designing Safe, Reliable Systems Using Scade. In Margaria, T., & Steffen, B. (Eds.), *Leveraging Applications of Formal Methods* (pp. 115–129). Berlin, Heidelberg: Springer Verlag. doi:10.1007/11925040_8

Abidin, M. S. Z., Rubiyah, Y., Marzuki, K., & Shamsuddin, M. A. (2002). Application of a model-based fault detection and diagnosis using parameter estimation and fuzzy inference to a DC servomotor. *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, (pp. 783-788).

Abrial, J. R. (1996). *The B-Book*. Cambridge University Press. doi:10.1017/CBO9780511624162

Abrial, J.-R. (2007). Theory Becoming Practice. In *Journal of Universal Computer Science* (pp. 619–628). Formal Methods.

Abul Masrur, M., Chen, Z., Murphey, & Y. (2010). Intelligent diagnosis of open and short circuit faults in electric drive inverter for real-time applications. *IET Power Electronics, 3*(2), 279-291.

Advantage Technical Consulting. (2002). *Review of the reliability of point motors and track circuits*.

Aeronautical Radio Inc. (2005). *ARINC 653, P1-2, Avionics Application Software Interface, Part 1, Required Services.*, Annapolis, MD: Aeronautical Radion Inc.

Aeronautical Radio Inc. (2009). *ARINC 664, P7-1, Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network.* Annapolis, Maryland: Aeronautical Radion Inc.

Akgun, I., Kandakoglu, A., & Ozok, A. F. (2010.), Fuzzy integrated vulnerability assessment model for critical facilities in combating the terrorism, in *Expert Systems with Applications, 37*.

Allegre, A. J., Verhille, J. N., Delarue, P., Chattot, E., & El-Fassi, S. (2010). Reduced-scale-power hardware in the loop simulation of an innovative subway. *IEEE Transactions on Industrial Electronics*, *57*(4), 1175–1185. doi:10.1109/TIE.2009.2029519

Allotta, B., Pisano, A., Pugi, L., & Usai, E. (2005, December), *VSC of a servo-actuated ATR90-type pantograph*. Paper presented at the 44th IEEE Conference on Decision Control, Seville, Spain.

Allotta, B., Pugi, L., & Bartolini, F. (2009a) "An active suspension system for railway pantographs: the T2006 prototype," *Proceedings of the IMechE, Part F: Journal of Rail and Rapit Transit, 223(* pp. 15-29)

Allotta, B., Pugi, L., & Bartolini, F. (2009b) Design and Testing of Innovative Pantographs: a general overview, *Proceedings of the International Seminar-Workshop on Power Transmission in High Speed Railway Systems*, Amiens, France, 4th December 2009

Allotta, B., Pugi, L., & Bartolini, F. (2010) *Mutual interaction of parallel connected induction motors on degraded adhesion conditions*, Proceedings of the 1st Joint International Conference on Multibody System Dynamics May 25-27, 2010, Lappeenranta, Finland

Allotta, B., Pugi, L., Malvezzi, M., Bartolini, F., & Cangioli, F. (2010) A scaled roller test rig for high-speed vehicles, *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility48*(1) 3 – 18

Allotta, B., Pugi, L., & Bartolini, F. (2008, October). Design and Experimental Results of an Active Suspension System for a High-Speed Pantograph. *IEEE/ASME Transactions on Mechatronics*, *13*(5). doi:10.1109/TMECH.2008.2002145

Almukhaizim, S., Petrov, P., & Orailoglu, A. (2001). Low-Cost, Software-Based Self-Test Methodologies for Performance Faults in Processor Control Subsystems. In *IEEE Conference on Custom Integrated Circuits* (pp. 263–266). San Diego, CA, USA: IEEE Computer Society Publications.

Alpe, S., Di Carlo, S., Prinetto, P., & Savino, A. (2008) Applying march tests to k-way set-associative cache memories. In *13th IEEE European Test Symposium* (pp. 77–83). Verbania, Italy: IEEE Computer Society Publications.

Amendola, A. M., di Maio, R., Iacobuzio, M. L., Poli, F., & Scalabrini, F. (2003): Lessons learned in designing and evaluating railway control systems, *Proc. of. Ninth IEEE International Workshop on Object-Oriented Real-Time Dependable Systems*, (pp. 355 – 358) 1-3 Oct. 2003

American Public Transportation Association (APTA). (2010). *Public Transportation Fact Book 2010, Appendix B: Transit Agency and Urbanized Area Operating Statistics*. Washington, DC: APTA.

Amin, M. (2002), Toward secure and resilient interdependent infrastructures, in *Journal of Infrastructure Systems, 8*, 67–75.

Annett, J. (2003): Hierarchical Task Analysis. In: LeBlanc Dobson, D. (Author); Hollnagel, E. (ed.): *Handbook of Cognitive Task Design*, (pp.17-35) Lawrence Erlbaum Associates

Antoni, M., & Ammad, N. (2008): Formal validation method and tools for French computerized railway interlocking systems, *4th IET International Conference on Railway Condition Monitoring*, 1-10

Apostolakis, G. E., & Lemon, D. M. (2005), A Screening Methodology for the Identification and Ranking of Infrastructure Vulnerabilities Due to Terrorism, in Risk Analysis, in *Risk Analysis, 25*, (2), 361-376.

Apostolakis, A., Gizopoulos, D., Psarakis, M., & Paschalis, A. (2009). Software-based self-testing of symmetric shared-memory multi-processors. *IEEE Transactions on Computers*, *58*(12), 1682–1694. doi:10.1109/TC.2009.118

Arabestani, S., Bitsch, F., & Gayen, J.-T. (2004). Precise Definition of the Single-Track Level Crossing in Radio-Based Operation in UML Notation and Specification of Safety Requirements. In *H. Ehrig, W. Damm, J. Desel, M. Große-Rhode, W. Reif, E. Schnieder, et al., Integration of Software Specification Techniques for Applications in Engineering* (pp. 119–144). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-27863-4_9

Arlat, J., & Moraes, R. (2011). *Collecting, Analyzing and Archiving Results from Fault Injection Experiments.* Paper presented at the 5th Latin-American Symposium on Dependable Computing.

Arlat, J., Crouzet, Y., Karlsson, J., Folkesson, P., Fuchs, E., & Leber, G. H. (2003, September). Comparison of Physical and Software Implemented Fault Injection Techniques. *IEEE Transactions on Computers*, *52*(9). doi:10.1109/TC.2003.1228509

Arnold, M., & Simeon, B. (2000). Pantograph and catenary dynamics: a benchmark problem and its numerical solution. *Journal of Applied Numerical Mathematics*, *34*(4), 345–362. doi:10.1016/S0168-9274(99)00038-0

ASIS International. (2003). *General Security Risk Assessment Guideline*. Alexandria, VA: ASIS International.

Association of American Railroads (AAR) Policy and Economics Department. (2010). *US Freight Railroad Statistics, November 2010*. AAR Publications.

Association of American Railroads (AAR). (2005) *Manual of Standards and Recommended Practices, Section K-Railway Electronics* AAR Publications.

Australian Level Crossing Assessment Model (ALCAM) Technical Committee. (2007). *Australian Level Crossing Assessment Model Technical Manual*. ALCAM.

Aven, T. (2007), A uni□ed framework for risk and vulnerability analysis covering both safety and security, in *Reliability Engineering and System Safety, 92*, Elsevier.

Bacherini, S., Fantechi, A., Tempestini, M., & Zingoni, N. (2006). A Story about Formal Methods Adoption by a Railway Signaling Manufacturer. In J. Misra, T. Nipkow, & E. Sekerinski (Eds.), *LNCS 4025: FM 2006: Formal Methods* (pp. 179-189), Hamilton, Canada. Berlin, Germany: Springer.

Baier, C., & Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.

Bajpai, S., Sachdeva, A., & Gupta, J. P. (2009). Security risk assessment: Applying the concepts of fuzzy logic. In *Journal of Hazardous Materials*. Elsevier. doi:10.1016/j.jhazmat.2009.08.078

Baker, P., Loh, S., & Weil, F. (2005). Model-driven engineering in a large industrial context — motorola case study. In Briand, L. and Williams, C., (eds) Model Driven Engineering Languages and Systems, volume 3713 of *Lecture Notes in Computer Science*, (pp 476–491). Berlin / Heidelberg: Springer. 10.1007/11557432_36.

Baldauf, W., Blaschko, R., Behr, W., Heine, C., & Kolbe, M. (2001) Development of an actively controlled, acoustically optimised single arm pantograph, *Proceedings of the World Congress of Railway Research WCRR 2001*, Cologne.

Baldini, F., Luise, P., Bagagli, R., Marcoccio, M., Senesi. M, (2010): The High Speed Railway infrastructure in Europe and its dependency on GSM-R wireless communication, *International Journal of Critical Infrastructure Protection*, 2010.

Balser, M., Reif, W., Schellhorn, G., Stenzel, K., & Thums, A. (2000). *Formal system development with KIV. Fundamental Approaches to Software Engineering*. Springer.

Banci, M., & Fantechi, A. (2005). Geographical vs. functional modelling by statecharts of interlocking systems. *Electronic Notes in Computer Science*, *133*, 3–19. doi:10.1016/j.entcs.2004.08.055

Banković, Z., Stepanović, B., Bojanić, S., & Nieto-Taladriz, O. (2007). Improving network security using genetic algorithm approach, in *Computers &. Electrical Engineering*, *33*, 438–451.

Banks, J. J., & Carson, S. Nelson, B. L. &,Nicol, D. M. (2000), *Discrete-Event System Simulation. (3rd Ed)*, Upper Saddle River, NJ: Prentice Hall.

Baraza, J. C., Gracia, J., Gil, D., & Gil, P. (2005, December). *Improvement of Fault Injection Techniques Based on VHDL Code Modification.* Paper presented at the Tenth IEEE International High-Level Design Validation and Test Workshop.

Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: Fundamentals, computing, design and application. *Journal of Microbiological Methods*, *43*, 3–31. doi:10.1016/S0167-7012(00)00201-3

Bayraktaroglu, I., Hunt, J., & Watkins, D. (2006). Cache resident functional microprocessor testing: Avoiding high speed IO issues. In IEEE *International Test Conference* (pp. 1-7), Austin, TX, USA: IEEE Computer Society Publications.

Becraft, W. R., & Lee, P. L. (1993). An integrated neural network/expert system approach for fault diagnosis. *Computers & Chemical Engineering*, *17*(10), 1001–1014. doi:10.1016/0098-1354(93)80081-W

Behm, P., Benoit, P., Faivre, A., & Meynadier, J. M. (1999). Météor: A successful application of B in a large project. *LNCS 1708: World Congress on Formal Methods in the Development of Computing Systems* (pp. 369–387). Toulouse, France. Berlin, Germany: Springer.

Ben-Ari, M. (2008). *Principles of the Spin Model Checker*. Springer Verlag.

Bengtsson, J., & Yi, W. (2004). Timed Automata: Semantics, Algorithms and Tools. In Rozenberg, R. W., & G., (eds) *Lecture Notes on Concurrency and Petri Nets*. Springer-Verlag.

Benso, A., Bosio, A., di Carlo, S., & Mariani, R. (2007, September). *A Functional Verification based Fault Injection Environment*. Paper presented at the 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.

Benso, A., Bosio, A., Prinetto, P., & Savino, A. (2006). An on-line software-based self-test framework for microprocessor cores. In *International Conference on Design and Test of Integrated Systems in Nanoscale Technology* (pp. 394–399). Tunis, Tunis: IEEE Computer Society Publications.

Benso, A., Di Carlo, S., Prinetto, P., Savino, A., & Scionti, A. (2008). Using ER models for microprocessor functional test coverage evaluation. In *11th International Biennial Baltic Electronics Conference* (pp. 139–142). Tallin, Estonia: IEEE Computer Society Publications.

Bepperling, S. (2008). *Validation of a semi-quantitative approach for risk assessement on railways* (in German), PhD thesis, Technical University of Brunswick

Berger, J., Middelraad, P., & Smith, A. J. (1993). EURIS, European railway interlocking specification. [Institution of Railway Signal Engineers.]. *Proceedings IRSE*, *93*, 70–82.

Bernardeschi, C., Fantechi, A., Gnesi, S., & Mongardi, G. (1996). Proving safety properties for embeddedcontrol systems. In *Procs. of Conference on Dependable Computing* (EDCC-2), *16*(440), 321-332. Springer-Verlag.

Bernardeschi, C., Fantechi, A., Gnesi, S., Larosa, S., Mongardi, G., & Romano, D. (1998). A formal verification environment for railway signaling system design. *Formal Methods in System Design*, *12*(2), 139–161. doi:10.1023/A:1008645826258

Bernardi, P., Bolzani, L., Manzone, A., Osella, M., Violante, M., & Sonza Reorda, M. (2006). Software-based on-line test of communication peripherals in processor-based systems for automotive applications. In *Seventh International Workshop on Microprocessor Test and Verification.* (pp. 3–8). Austin, TX, USA: IEEE Computer Society Publications.

Berry, G. (2003). *The Effectiveness of Synchronous Languages for the Development of Safety-Critical Systems*. Retrieved April 3, 2011 from http://www.esterel-technologies.com/DO-178B/files/The-Effectiveness-of-Synchronous-Languages-for-the-Development-of-Safety-Critical-Systems.pdf

Berthomieu, B., Ribet, P.-O., & Vernadat, F. (2004). The tool TINA - Construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research.*

Beyer, D., Henzinger, T. A., Jhala, R., & Majumdar, R. (2007). The Software Model Checker Blast. *International Journal on Software Tools for Technology Transfer*, *9*(5-6), 505–525. doi:10.1007/s10009-007-0044-z

Bias, R. G., & Mayhew, D. J. (Eds.). (1994). *Cost-Justifying Usability -An Update for the Internet Age*. San Francisco, CA: Morgan Kaufmann Publishers.

Biere, A., Cimatti, A., Clarke, E. M., & Zhu, Y. (1999) Symbolic model checking without BDDs. *LNCS 1579: Tools and Algorithms for Construction and Analysis of Systems*, (pp. 193-207). Amsterdam, The Netherlands. Berlin, Germany: Springer.

Bitsch, F. (2001). Safety Patterns - The Key to Formal Specification of Safety Requirements. *Proceedings of the 20th International Conference on Computer* (pp. 176-189). Berlin / Heidelberg: Springer Verlag.

Blanc, S., Bonastre, A., & Gil, P. G. (2009). Dependability assessment of by-wire control systems using fault injection. *Journal of Systems Architecture*, ▪▪▪, 55.

Bochot, T., Virelizier, P., Waeselynck, H., & Wiels, V. (2009). Model checking flight control systems: The Airbus experience. *ICSE Companion*, *2009*, 18–27.

Boiteux, M. (1986) Le problème de l'adèrence en freinage [The problem of adhesion in braking], *Revue gènèrale des chemins de fer*,[General review of the railways] (pp. 59–72.) Fèvrier

Bondy, J. A., & Murty, U. S. R. (1980). *Graph Theory with Applications*. New York, NY: North-Holland.

Bonta, D., Festila, R., & Tulbure, V. (2006, May). *The problem of speed measurements in the slip-slide control for electric railway tractions*. Paper presented at the IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania.

Bordas, C., Dufour, C., & Rudloff, O. (2009). *A 3-level neutral-clamped inverter model with natural switching mode support for the real-time simulation of variable speed drives*. Paper presented at the 8th International Symposium on Advanced Electromechanical Motion Systems, Lille, France.

Bose, B. K. (Ed.). (2006). *Power Electronics and Motor Drives-Advances and Trends*. Burlington, MA: Academic Press.

Bowles, J. (2003) *An Assessment of RPN Prioritization in a Failure Modes Effects and Criticality Analysis*, Proc. RAMS2003, Tampa, 2003

Bozzano, M., & Villafiorita, A. (2010). *Design and Safety Assessment of Critical Systems.* CRC Press (Taylor and Francis), an Auerbach Book.

Bozzano, M., Cimatti, A., Roveri, M., Katoen, J.-P., Nguyen, V. Y., & Noll, T. (2009). Codesign of dependable systems: A component-based modeling language. In *7th IEEE/ACM International Conference on Formal Methods and Models for Co-Design, MEMOCODE* (pp. 121 -130). IEEE.

Bozzano, M., Cimatti, A., Katoen, J.-P., Nguyen, V., Noll, T., & Roveri, M. (2009). The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In Buth, B., Rabe, G., & Seyfarth, T. (Eds.), *Computer Safety, Reliability, and Security* (pp. 173–186). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-642-04468-7_15

Bozzano, M., & Villafiorita, A. (2007). The FSAP/NuSMV-SA Safety Analysis Platform. In *International Journal on Software Tools for Technology Transfer (STTT)* (pp. 5–24). Berlin, Heidelberg: Springer Verlag.

Braband, J. (2005). *Risk analyses in railway automation*. Hamburg: Eurailpress. (in German)

Braband, J. (2011). On the Justification of a Risk Matrix for Technical Systems in European Railways. In Schnieder, E. (Ed.), *FORMS/FORMAT 2010* (pp. 237–288). Springer. doi:10.1007/978-3-642-14261-1_19

Brahme, D., & Abraham, J. A. (1984). Functional testing of microprocessors. *IEEE Transactions on Computers*, C-*33*(6), 475–485. doi:10.1109/TC.1984.1676471

Bruno, O., Landi, A., Papi, M., & Sani, L. (2001). Phototube sensor for monitoring the quality of current collection on overhead electrified railway. *Proceedings of the Institution of Mechanical Engineers. Part F, Journal of Rail and Rapid Transit*, *215*(3), 231–241. doi:10.1243/0954409011531549

Bruns, G., & Anderson, S. (1993). Validating Safety Models with Fault Trees. In J. Górski, *SafeComp '93: 12th International Conference on Computer Safety, Reliability, and Security* (pp. 21-30). Springer-Verlag.

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, *C-35*(8), 677–691. doi:10.1109/TC.1986.1676819

Bryant, R. E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE TransactionsOn Computers*, C-35(8), IEEE.

Boralv, A. & Stålmarck, G. (1999). Formal verification in railways. In Hinchey, M., & Bowen, J. (Eds.), *Industrial-Strength Formal Methods in Practice*. Springer-Verlag.

Bucca, G., & Collina, A. (2009). A procedure for the wear prediction of collector strip and contact wire in pantograph-catenary system. *Wear*, *266*(1-2), 46–59. doi:10.1016/j.wear.2008.05.006

Bucca, G., Collina, A., Manigrasso, R., Mapelli, F., & Tarsitano, D. (2010). Methodology for correlating the quality of the pantograph-catenary contact with the harmonic content of the current collected. A case of multiple current collection. *Ingegneria Ferroviaria.*, *3*, 211–237.

Budinsky, F., Steinberg, D., Merks, E., Ellersick, R., & Grose, T. (2003). *Eclipse Modeling Framework*. Addison Wesley Professional.

Burch, J. R., Clarke, E., McMillan, K., Dill, D., & Hwang, L. (1992). Symbolic model checking1020 states and beyond. *Information and Computation*- Special issue: Selections from 1990IEEE symposium on logic in computer science, *98*(2), 142-170.

Burch, J., Clarke, E., & Long, D. (1991). Symbolic model checking with partitioned transition relations. In *Int. Conf. on Very Large Scale Integration*.

Busco, B., Marino, P., Porzio, M., Schiavo, R., & Vasca, F. (2003). Digital control and simulation for power electronic apparatus in dual voltage railway locomotive. *IEEE Transactions on Power Electronics*, *18*(5), 1146–1157. doi:10.1109/TPEL.2003.816198

Butler, K. L. (1996). An expert system-based framework for an incipient failure detection and predictive maintenance system. *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems*, (pp. 321-326).

Carreira, J., Madeira, H., & Silva, J. G. (1995, September). *Xception: Software Fault Injection and Monitoring in Processor Functional Units.* Paper presented at the 5th IFIP Working Conference on Dependable Computing for Critical Applications.

Cassandras, C., & Lafortune, S. (2008). *Introduction to Discrete Event Systems* (2nd ed.). Springer Science. doi:10.1007/978-0-387-68612-7

Cavada, R., Cimatti, A., Mariotti, A., Mattarei, C., Micheli, A., Mover, S., et al. Susi, A. & Tonetta S. (2009) EuRail-Check: Tool Support for Requirements Validation. *ASE 2009* (pp. 665-667), Auckland, New Zealand. Washington D.C.: IEEE Computer Society

Cavda, R., Cimatti, A., Jochim, C. A., Keighren, G., Olivetti, E., Pistore, M., et al. (2110). *NuSMV 2.5 User Manual*. Retrieved March, 7, 2011 from http://nusmv.irst.itc.it.

CBMC. (n.d.), Retrieved from: http://www.cprover.org/cbmc/

CENELEC - European Committee for Electrotechnical Standardization. (1999). 50126-1:1999 *Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process.* CENELEC.

CENELEC (1997) E*N 50126 Railway applications –The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)*

CENELEC (1999). EN 50126-1 Railway applications - The specification and demonstration of Reliability, *Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process. Corrigendum 2006*.

CENELEC (2010) prEN 15380 Part 4: *Railway applications – Classification system for rail vehicles – Function groups*

CENELEC EN 50128:2002. (2002) *Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems*.

CENELEC EN 50129:2003. (2003) *Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signalling*.

CENELEC EN 50159-1:2001. (2001) *Railway applications. Communication, signalling and processing systems. Safety related communication in closed transmission systems". EN 50159-2:2001 "Railway applications. Communication, signalling and processing systems. Safety related communication in open transmission system"*.

CENELEC EN 50367 (May 2010) *Railway applications - Current collection systems - Technical criteria for the interaction between pantograph and overhead line (to achieve free access) - Incorporating corrigendum*

CENELEC. (2001). *EN 50128, Railway Applications - Communications*. Signaling and Processing Systems - Software for Railway Control and Protection Systems.

CENELEC. (2011). *Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems (EN 50128:2011).*

CENELEC-EN 50206-1 (2010) *Railway applications - Rolling stock - Pantographs: Characteristics and tests - Part 1: Pantographs for main line vehicles* CENELEC

Chen, C.-H., Wei, C.-K., Lu, T.-H., & Gao, H.-W. (2007). Software- based self-testing with multiple-level abstractions for soft processor cores. *IEEE Transactions on Very Large Scale Integration (VLSI)*. *Systems*, *15*(5), 505–517.

Cheung, J. T.-Y., & Stephanopoulos, G. (1990). Representation of process trends, part I: A formal representation framework. *Computers & Chemical Engineering*, *14*(4-5), 495–510. doi:10.1016/0098-1354(90)87023-I

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking. *Proc. International Conference on Computer-Aided Verification (CAV 2002)*. Copenhagen, Denmark: Springer.

Cimatti, A., Giunchiglia, F., Mongardi, G., Romano, D., Torielli, F., & Traverso, P. (1998). Formal verification of a railway interlocking system using model checking. *Formal Aspects of Computing*, *10*(4), 361–380. doi:10.1007/s001650050022

Clarke, E.M., Grumberg, O. & Peled. D. (1999) *Model Checking*. MIT PRESS.

Clarke, E., Grumberg, O., Jha, S., Lu, Y., & Veith, H. (2000). Counterexample-Guided Abstraction Refinement. In Emerson, E., & Sistla, A. (Eds.), *Computer Aided Verification* (pp. 154–169). Berlin, Heidelberg: Springer Verlag. doi:10.1007/10722167_15

Clarke, E., Grumberg, O., & Peled, D. (2000). *Model Checking*. MIT Press.

Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., & Meseguer, J. (2007). *All About Maude - A High-Performance Logical Framework*. Springer Verlag.

Cleaveland, R., Luettgen, G., & Natarajan, V. (1996). Modeling and verifying distributed systems using priorities: A case study. In *Procs. of Int. Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'96), volume 1055 of LNCS,(* pp. 287-297). Springer-Verlag.

Cocci, G., Malvezzi, M., Palazzolo, A., Presciani, P., Pugi, L., & Violani, M. (2006) *Braking Performance Monitoring in Service for the Validation of the Safety Margins used for the Definition of Braking Curves of ATP/ATC Systems*, World Congress on Railway Research 5-7June 2006 Montreal

Codetta-Raiteri, D. (2005). The Conversion of Dynamic Fault Trees to Stochastic Petri Nets, as a case of Graph Transformation. In *Electronic Notes Theoretical Computer Science* (pp. 45-60).

Collina, A., Facchinetti, A., Fossati, F., & Resta, F. (2004). Hardware in the loop test-rig for identification and control application on high speed pantographs. *Shock and Vibration*, *11*(3-4), 445–456.

Commission, E. U. (2005). *Council Directive 2005/65/EC of October 26th 2005 on enhancing port security*. Official Journal of the European Union.

Commission, E. U. (2008). *Council Directive 2008/114/EC of December 8th 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection*. Official Journal of the European Union.

Commission, E. U. (2010). *Council Regulation 2010/186 of March 4th 2010 laying down detailed measures for the implementation of the common basic standards on aviation security*. Official Journal of the European Union.

Coppit, D., Sullivan, K., & Dugan, J. (2000). Formal Semantics of Models for Computational Engineering: A Case Study on Dynamic Fault Trees. In *International Symposium on Software Reliability Engineering*. IEEE.

Corbrion, C., T. Ditchi, T., Hole, S., Carreel, E., & Lewiner, J.,(2001) A broad beam Doppler speed sensor for automotive applications, *MCB University Press Sensor Review 21*(1). 28-32

Corno, F., Sanchez, E., Sonza Reorda, M., & Squillero, G. (2004). Automatic test program generation: a case study. *IEEE Design & Test of Computers*, *21*(2), 102–109. doi:10.1109/MDT.2004.1277902

Cousot, P., & Cousot, R. (1977). Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages* (pp.238-353), Los Angeles, CA. New York, NY: ACM Press.

Cousot, P., Cousot, R., Feret, J., Mauborne, L., Miné, A., & Rival, X. (2009). Why does Astrée scale up? *Formal Methods in System Design*, *35*(3), 229–264. doi:10.1007/s10703-009-0089-6

Damm, W., Josko, B., Hungar, H., & Pnueli, A. (1998). A Compositional Real-time Semantics of STATEMATE Designs. *COMPOS'97, Volume 1536 of LNCS* (pp. 186-238). Berlin / Heidelberg: Springer.

DaSilva, C., Dehbonei, Y., & Mejia, F. (1993) Formal specification in the development of industrial applications: Subway speed control system. *5th IFIP Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE'92)*, (pp. 199–213). Perros-Guirec, France. Amsterdam, The Netherlands: North-Holland.

del Portillo, J., Osinalde, M., Sukia, E., Sancho, I., Medizabal, J., & Meléndez, J. (2008): Characterization of the EM environment of railway spot communication systems. In *Symposium on Electromagnetic Compatibility, 2008. EMC 2008*. IEEE International.

Department of Transport and Main Roads Queensland - Rail Safety Regulation Branch. (2010). *Rail level crossing occurrence statistics*. Brisbane.

Department of Transport NSW. (2010). *The Australian Level Crossing Assessment Model: ALCAM in Detail.*

Deutsch, A. (2004) Static verification of dynamic properties. Polyspace white paper.

Dhahbi, S., Abbas-Turki, A., Hayat, S., & El Moudni, A. (2011). *Study of the high-speed trains positioning system: European signaling system ERTMS / ETCS*. Paper presented at the 4th International Conference on Logistics.

Di Carlo, S., & Prinetto, P. (2010). Models in Memory Testing, From functional testing to defect-based testing. In Wunderlich, H.-J. (Ed.), *Models in Hardware Testing* (pp. 157–185). Springer DEU.

Di Carlo, S., Prinetto, P., & Savino, A. (2011). Software-based self-test of set-associative cache memories. *IEEE Transactions on Computers*, *60*(7), 1030–1044. doi:10.1109/TC.2010.166

Di Febbraro, A., & Sacco, N. (2010*), A Petri-Net based approach for the interdependence analysis of Critical Infrastructures in transportation networks*, in *Proceedings of 12th World Conference on Transportation Research*. COTA.

Di Febbraro, A., Papa, F., & Sacco, N. (2010), A Tool for Risk Analysis and Protection Design of Railway Infrastructures, in *Proceedings of 89th TRB Annual Meeting*. TRB.

Di Pietro, C., Vasca, F., Iannelli, L., & Oliviero, F. (2010, October), *Decentralized synchornization of parallel inverters for train auxiliaries*. Paper presented at the International Conference on Electrical Systems for Aircraft, Railway and Ship Propulsion, Bologna, Italy.

Di Tommaso, P., Flammini, F., Lazzaro, A., Pellecchia, R., & Sanseviero, A. (2005, October). *The simulation of anomalies in the functional testing of the ERTMS/ETCS trackside system*. Paper presented at 9th IEEE International Symposium on High-Assurance Systems Engineering, Heidelberg, Germany.

DIN (2011) *Semi-quantitative processes for risk analysis of technical functions in railway* signalling (in German), DIN V VDE V 0831-101

Dingel, J., Diskin, Z., & Zito, A. (2008). Understanding and improving UML package merge. *Software and Systems Modeling*, *7*(4), 443–467. doi:10.1007/s10270-007-0073-9

Directive (2009*) 2008/57/EC on the interoperability of the rail system within the Community Technical Specification for Interoperability*, version EN02 del 17.12.2009

Dommel, H. W. (1969). Digital computer solution of electromagnetic transients in single- and multiphase networks. *IEEE Transactions on Power Apparatus and Systems*, *88*(4), 388–399. doi:10.1109/TPAS.1969.292459

Dufour, C., Dumur, G., Paquin, J. N., & Belanger, J. (2008, June). *A PC-based hardware in the loop simulation for the integration testing of modern train and ship propulsion systems*. Paper presented at the 39th IEEE Power Electronics Specialists Conference, Island of Rhodes, Greece.

Dufour, C., Bélanger, J., & Abourida, S. (2003). Accurate simulation of a 6-pulse inverter with real time event compensation in ARTEMIS. *Mathematics and Computers in Simulation*, *63*(3-5), 161–172. doi:10.1016/S0378-4754(03)00072-7

Dufour, C., Mahseredjian, J., & Bélanger, J. (2011). A combined state-space nodal method for the simulation of power system transients. *IEEE Transactions on Power Delivery*, *26*(2), 928–935. doi:10.1109/TPWRD.2010.2090364

Dumas, J. S., & Redish, J. C. (1999). *A Practical Guide to Usability Testing*. Intelect.

Dutertre, J., Fourniery, J., Mirbaha, A., Naccachez, D., Rigaud, J., Robissony, B., & Triay, A. (2011). *Review of Fault Injection Mechanisms and Consequences on Countermeasures Design*. Paper presented at the 6th International Conference on Design & Technology of Integrated Systems in Nanoscale Era.

Eames, D. P., & Moffett, J. (1999), The Integration of Safety and Security Requirements, in *Proceedings of the 18th International Conference on Computer Safety, Reliability and Security,* (, pp. 468-480) Springer Verlag

EC (2009) *Regulation No. 352/2009 of 24 April 2009 on the adoption of a common safety method on risk evaluation and assessment* as referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council

*EC 352/2009 (2009).*COMMISSION REGULATION (EC) No 352/2009 of 24 April 2009 on the adoption of a common safety method on risk evaluation and assessment as referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council

Edelkamp, S., & Sulewski, D. (2010). Efficient explicit-state model checking on general purpose graphics processors. In *Proceedings of the 17th international SPIN conference on Model checking software* (pp. 106-123). Enschede, The Netherlands: Springer Verlag.

Efkemann, C., & Peleska, J. (2011). *Model-Based Testing for the Second Generation of Integrated Modular Avionics*. To Appear in Proceedings of the A-MOST 2011.

Eisenecker, U. W., & Czarnecki, K. (2000). *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley.

Eisner, C. (1999). Using symbolic model checking to verify the railway stations of Hoorn-Kersenboogerd and Heerhugowaard. In *Proc. of Conf. on Correct Hardware Design and Verification Methods (CHARME'99), volume 1703 of LNCS.* Springer-Verlag.

Eisner, C. (2002). Using symbolic CTL model checking to verify the railway stations of Hoorn- Kersenboogerd and Heerhugowaard. *Software Tools for Technology Transfer*, *4*(1), 107–124. doi:10.1007/s100090100063

Elvik, R., Høye, A., Vaa, T., & Sørensen, M. (2009). *The Handbook of Road Safety Measures* (2nd ed.). Bingley, UK: Emerald Group Publishing Limited.

EN 50126 (1999) Railway applications - The specification and demonstration of reliability, availability, maintainability and safety (RAMS)

EN50121. (2006). *Railway applications - Electromagnetic compatibility – Parts 1, 2, 3-1, 3-2, 4 and 5*. CENELEC.

EN50215. (1999). *Railway applications. Testing of rolling stock after completion of construction and before entry into service*. CENELEC.

EN50238. (2003). *Railway applications, Compatibility between rolling stock and train detection systems*. CENELEC.

EN50338. (2001). *Railway applications. Power supply and rolling stock. Technical criteria for the coordination between power supply (substation) and rolling stock to achieve interoperability*. CENELEC.

ERA EMC Report. (2010). *67575_ERA_EMC_Final_Report - Study to collect and document rules, processes and procedures to verify the electromagnetic compatibility of railway vehicles in member states of the European rail area, for ERA*. Lloyd's Register Group.

Ericson, C. A. (2005). *Hazard Analysis Techniques for System Safety*. Hoboken, NJ: John Wiley & Sons, Inc. doi:10.1002/0471739421

ERRI B 126/RP 18 (2000) *Braking Problems, Dynamometers for internal approval of friction materials*.

ERTMS. (n.d.) Retrieved from: http://www.ertms.com.

Esposito, R., Lazzaro, A., Marmo, P., & Sanseviero, A. (2003) Formal verification of ERTMS Euroradio safety critical protocol. *4th Symposium on Formal Methods for Railway Operation and Control Systems (FORMS'03),* Budapest. L'Harmattan Hongrie.

European Commission (EC) (2010) *2010/79/EC Commission Decision of 19 October 2009 amending Decisions 2006/679/EC and 2006/860/EC as regards technical specifications for interoperability relating to subsystems of the trans-European conventional and high-speed rail systems* (notified under document C(2009) 7787)

European Commission. (2006). *Sustainable Surface Transport Research Technological Development and Integration*.

European Commission. (2010): *White Paper - European transport policy for 2010: time to decide*, ISBN 92-894-0341-1

European Committee for Electrotechnical Standardization. (2000). *EN 50126 - Railway applications - Communnications, signalling and processing systems – Specification and demonstration of reliability, availability, maintainability and safety (RAMS)*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001). *EN 50128 - Railway applications - Communnications, signalling and processing systems - Software for railway control and protection systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001). *EN 50159-1 - Railway applications - Communnications, signalling and processing systems – Safety-related communication in closed transmission systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2001). *EN 50159-2 - Railway applications - Communnications, signalling and processing systems – Safety-related communication in open transmission systems*. Brussels: CENELEC.

European Committee for Electrotechnical Standardization. (2003). *EN 50129 - Railway applications - Communnications, signalling and processing systems – Safety-relevant electronic systems for signaling technology*. Brussels: CENELEC.

European Railway Agency. (2007). *ERTMS/ETCS Functional Requirements Speci□cation FRS*. (Version 5.0)

European Union (EU) Eurostat. (2010) Regional *Transport Statistics- Railway Transport Measurement* - retrieved from http://epp.eurostat.ec.europa.eu/portal/page/portal/transport/introduction

Facchinetti, A., & Mauri, M. (2009). Hardware-in-the-Loop Overhead Line Emulator for Active Pantograph Testing. *IEEE Transactions on Industrial Electronics*, *56*(10). doi:10.1109/TIE.2009.2023632

Faiveley (2000) *Anti-skid system ANG 06/98 30/08/00 12:2* Datasheet retrieved from: http://www.faiveley.com/uk

Fantechi, A., Fokkink, W., & Morzenti, A. (2011). Some Trends in Formal Methods Applications to Railway Signaling. In Gnesi, S., & Margaria, T. (Eds.), *Formal Methods for Industrial Critical Systems: A Survey of Applications*. Wiley-IEEE Computer Society Press.

Farail, P., Gaufillet, P., Peres, F., Bodeveix, J.-P., Filali, M., Berthomieu, B., et al. (2008). FIACRE: an intermediate language for model verification in the TOPCASED environment. *European Congress on Embedded Real-Time Software (ERTS)*. SEE.

Farrow, S. (2004), Using Risk Assessment, Benefit-Cost Analysis, and Real Options to Implement a Precautionary Principle, in *Risk Analysis, 24*.

Favo. F., Alterisio G., & Illibato G.(2011): La certificazione in esercizio delle linee SCMT, *Tecnica Professionale [Professional Techniques]*, May 2011

Fazeli, M., Farivar, R., & Miremadi, S. (2005). A software-based concurrent error detection technique for PowerPC processor-based embedded systems. In *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems* (pp. 266–274). Monterey, CA, USA: IEEE Computer Society Publications.

Federal Railroad Administration (FRA) (2004) *Benefits and Costs of Positive Train Control. Report in Response to Request of Appropriations Committees. August 2004*

Federal Railroad Administration (FRA) (2006) *U.S. DOT/FRA - Letter Approving BNSF's Product Safety Plan Ver. 2.1 dated 21 December, 2006 Docket ID: FRA-2006-23687*

Federal Railroad Administration (FRA) Office of Railroad Development. (2009). *North American Joint Positive Train Control (NAJPTC)*. Project. Research Results.

Federal Railroad Administration (FRA) Office of Safety Analysis. (2010), *Operational Data Tables, Table 1.02* retrieved from http://safetydata.fra.dot.gov/OfficeofSafety

Federal Railroad Administration (FRA). (1994). *Railroad Communications and Train Control*. FRA.

Federal Railroad Administration (FRA). (2005). *FRA Emergency Order Number 24- Emergency Order Requiring Special Handling, Instruction, and Testing of Railroad Operating Rules Pertaining to Hand Operated Main Track Switches*. FRA.

Fenelon, P., McDermid, J., Nicholson, A., & Pumfrey, D. (1995). Experience with the application of HAZOP to computer-based systems. *Proceedings of the 10th Annual Conference on Computer Assurance.* Gaithersburg, MD: IEEE.

Ferrari, A., Fantechi, A., Tempestini, M., & Zingoni, N. (2009). Modelling Guidelines for Code Generation in the Railway Signaling Context. *Proceedings of 1st NASA Formal Methods Symphosium (NFM)* (pp 166-170). Moffet Field, CA, USA.

Ferrari, A., Grasso, D., Magnani, G., Fantechi, A., & Tempestini, M. (2010). The Metro Rio ATP case study. S. Kowalewski, & M. Roveri (Eds.), *LNCS 6371: 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS 2010)*, (pp. 1-16)Antwerp, Belgium. Berlin, Germany: Springer.

Ferrari, A., Magnani, G., Grasso, D., & Fantechi, A. (2011). Model checking interlocking control tables. In Schnieder, E. & Tarnai, G. (Eds.), *Proceedings of Conference on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 2010), 2*, (pp. 107 – 115). Springer-Verlag.

Ferrari, A., Magnani, G., Grasso, D., Fantechi, A., & Tempestini, M. (2011). Adoption of Model-Based Testing and Abstract Interpretation by a Railway Signaling Manufacturer. *International Journal of Embedded and Real-Time Communication Systems*, *2*(2), 42–61. doi:10.4018/jertcs.2011040103

Feuser, J., & Peleska, J. (2010). Security in Open Model Software with Hardware Virtualisation – The Railway Control Systems perspective. *Electronic Communications of the EASST, 33: Foundations and Techniques for Open Source Software Certification.*

Feuser, J., & Peleska, J. (2011). Dependability in Open Model Software with Hardware Virtualisation – The Railway Control System Perspective. Submitted to *Science of Computer Programming.*

Fidalgo, A. V., Alves, G. R., & Ferreira, J. M. (2006). *Real Time Fault Injection Using Enhanced OCD – A Performance Analysis*. Paper presented at the 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems.

Fink, C. N. Y. (2003), Antiterrorism Security and Surface Transportation Systems - Review of Case Studies and Current Tactics, in *Journal of the Transportation Research Board, 1822*, 9–17.

Fringuelli, B., Lamma, E., Mello, P., & Santocchia, G. (1992). Knowledge-based technology for controlling railway stations. *IEEE Intelligent Systems*, *7*(6), 45–52.

Gagnon, E. (1998). *SableCC, an object-oriented compiler framework*. PhD thesis, McGill University.

Gansner, E. R. (2003). *Drawing graphs with GraphViz. Technical report*. Murray Hill, NJ, USA: AT&T Bell Laboratories.

Garcia, M. L. (2001). *The design and evaluation of physical protection systems*. Elsevier.

Garrick, B. J., Hallb, J. E., Kilgerc, M., McDonaldd, J. C., O'Toolee, T., Probstf, P. S., et al. (2004*), Confronting the risks of terrorism: making the right decisions,* in *Reliability Engineering & System Safety, 86*, 129-176.

Gayme, D., Menon, S., Ball, C., Mukavetz, D., & Nwadiogbu, E. (2003). Fault detection and diagnosis in turbine engines using fuzzy logic. *NAFIPS 2003: 22nd International Conference of the North American Fuzzy Information Processing Society*, (pp. 341-346).

Geist, D., & Beer, I. (1994). Efficient model checking by automated ordering of transition relation. In Dill, D., (Ed.), *Proc. of Int. Conference on Computer-Aided Verification* (CAV'94), *volume 818 of LNCS*, (pp. 299-310.) Springer-Verlag.

Giese, H., & Tichy, M. (2006). Component-Based Hazard Analysis: Optimal Designs, Product Lines, and Online-Reconfiguration. In Górski, J., *Proc. of the 25th International Conference on Computer Safety, Security and Reliability (SAFECOMP)* (pp. 156-169). Gdansk, Poland: Springer Verlag.

Giese, H., Tichy, M., & Schilling, D. (2004). Compositional Hazard Analysis of UML Components and Deployment Models. In M. Heisel, P. Liggesmeyer, & S. Wittmann, *Proc. of the 23rd International Conference on Computer Safety, Reliability and Security (SAFECOMP)* (pp. 166-179). Potsdam, Germany: Springer Verlag.

Gil, M., Multer, J., & Yeh, M. (2007). *Effects of Active Warning Reliability on Motorist Compliance at Highway-Railroad Grade Crossings*.

Giugno L., Luise M., Bagagli E., Giannini M., Senesi F., Malangone R., & Caronti D.(2008): Valutazione dei fattori di rischio indisponibilità nell'uso della rete radio GSM-R per applicazioni ferroviarie italiane ad alta velocità/alta capacità (AV/AC) [Assessment of risk factors is unavailable in the use of the radio network for GSM-R applications Italian railway high speed / high capacity], *Ingegneria Ferroviaria [Railway Engineering],* January 2008

Gizopoulos, D., Paschalis, A., & Zorian, Y. (2004). *Embedded Processor-Based Self-Test*. Springer Press.

Gizopoulos, D., Psarakis, M., Hatzimihail, M., Maniatakos, M., Paschalis, A., Raghunathan, A., & Ravi, S. (2008). Systematic software-based self-test for pipelined processors. *IEEE Transactions on Very Large Scale Integration (VLSI)*. *Systems*, *16*(11), 1441–1453.

Glover, I., & Grant, P. (2003) *Digital Communications (2nd ed)*ISBN978-0130893994, Upper Saddle River, NJ: Prentice Hall

Gnesi, S., Lenzini, G., Latella, D., Abbaneo, C., Amendola, A., & Marmo, P. (2000). An automatic SPIN validation of a safety critical railway control system. In *Procs. of IEEE Conference on Dependable Systems and Networks*, (pp. 119-124.) IEEE Computer Society Press.

Goodall, R. M. (2011, June). *Control for railways – active suspensions and other opportunities*. Paper presented at the 19th Mediterranean Conference on Control and Automation, Corfu, Greece.

Goos, G., & Zimmermann, W. (1999). Verification of compilers. In Olderog, E.-R., & Steffen, B. (Eds.), *Correct System Design, Recent Insight and Advances* (pp. 201–230). Springer-Verlag.

Górski, J. (1994). *Extending safety analysis techniques with formal semantics. Technology and Assessment of Safety Ciritcal Systems* (pp. 147–163). London: Springer Verlag.

Groote, J. F., Koorn, J. W. C., & van Vlijmen, S. F. M. (1995). The safety guaranteeing system at station Hoorn-Kersenboogerd. In *Proceedings 10th IEEE Conference on Computer Assurance* (COMPASS95), (pp. 131-150). IEEE Computer Society Press.

Grötker, T., Liao, S., Martin, G., & Swan, S. (2002). *System Design with SystemC*. Kluwer Academic Publishers.

Güdemann, M., Ortmeier, F., & Reif, W. (2007). Using Deductive Cause Consequence Analysis (DCCA) with SCADE. *Proceedings of SAFECOMP 2007.* Springer LNCS 4680.

Güdemann, M., Ortmeier, F., & Reif, W. (2008). Computing Ordered Minimal Critical Sets. *Proceedings of the 7th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 08)*.

Guglielmino, E. (November, 2004) Flexible Waveform Generation Accomplishes Safe Braking, *Evaluation Engineering*, Retrieved from: http://www.evaluationengineering.com

Gurevich, Y. (1995). Evolving Algebras 1993: Lipari Guide. In Börger, E. (Ed.), *Specification andValidation Methods*. Oxford University Press.

Gurumurthy, S., Vasudevan, S., & Abraham, J. A. (2006). Automatic generation of instruction sequences targeting hard-to-detect structural faults in a processor. In *IEEE International Test Conference* (pp. 1–9). Austin, TX, USA: IEEE Computer Society Publications.

Hammerl, M. (2011) *Analyse der menschlichen Einflussfaktoren und Zuverlässigkeit im Eisenbahnverkehr. (Analysis of human factors and reliability in railways)*. PhD thesis. TU Braunschweig, Germany.

Hänsel, F., Poliak, J., Slovák, R., & Schnieder, E. (2004). Reference Case Study "Traffic Control Systems" for Comparison and Validation of Formal Specifications Using a Railway Model Demonstrator. In Ehrig, H., Damm, W., Desel, J., Große-Rhode, M., Reif, W., & Schnieder, E. (Eds.), *Integration of Software Specification Techniques for Applications in Engineering* (pp. 96–118). Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-27863-4_8

Hansen, K. M., Ravn, A. P., & Stavridou, V. (1994). *From safety analysis to formal specification. ProCoS II document* [ID/DTH KMH 1/1]. Technical University of Denmark.

Hansen, K., Ravn, A., & Stavridou, V. (1998). From safety analysis to software requirements. [IEEE.]. *Transactions on Software Engineering*, *24*(7), 573–584. doi:10.1109/32.708570

Harakawa, M., Dufour, C., Nishimura, S., & Nagano, T. (2009, September). *Real-time simulation of a PMSM drive in faulty modes with validation against an actual drive system*. Paper presented at the 13th European Conference on Power Electronics and Applications, Barcelona, Spain.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231–274. doi:10.1016/0167-6423(87)90035-9

Harel, D., & Naamad, A. (1996). *The STATEMATE semantics of statecharts. Transactions on Software Engineering and Methodology* (pp. 293–333). New York, NY, USA: ACM.

Hartong, M., Goel, R., & Wijesekera, D. (2008), Security and the US rail infrastructure, in *International Journal of Critical Infrastructure Protection, 1*, 15-28.

Hase, K. R. (2009). openETCS - Ein Vorschlag zur Kostensenkung und Beschleunigung der ETCS-Migration. *Signal+Draht 10*(10).

Hase, K. R. (2009). openETCS - Open Source Software für ETCS-Fahrzeugausrüstung. *Signal+Draht*(12)12.

Hase, K. R. (2011). "Open Proof" for Railway Safety Software - A Potential Way-Out of Vendor Lock-in Advancing to Standardization, Transparency, and Software Security. In Schnieder, E., & Tarnai, G. (Eds.), *FORMS/FORMAT 2010 Formal Methods for Automation and Safety in Railway and Automotive Systems* (pp. 4–34). Berlin, Heidelberg: Springer-Verlag.

Havelund, K., Lowry, M., Pecheur, C., Penix, J., Visser, W., & White, J. (2000). Formal Analysis of the Remote Agent Before and After Flight. In *The Fifth NASA Langley Formal Methods Workshop.* Virginia.

Haxthausen, A. E., & Peleska, J. (2007). A domain-oriented, model-based approach for construction and verification of railway control systems. In *Formal Methods and Hybrid Real-Time Systems*, (pp 320–348.)

Haxthausen, A. E. (2010). *An Introduction to Formal Methods for the Development of Safety-critical Applications. Kgs*. Denmark: Lyngby.

Haxthausen, A. E., Peleska, J., & Kinder, S. (2011). A formal approach for the construction and verification of railway control systems. *Formal Aspects of Computing*, *23*(2), 191–219. doi:10.1007/s00165-009-0143-6

Hayasaka, T., Shimizu, M., & Nezu, K. (2009). Development of Contact-Loss Measuring System Using Ultraviolet Ray Detection Development of Contact-Loss Measuring System Using Ultraviolet Ray Detection. *Quarterly Report of*, *RTRI50*(3), 131–136.

Heinsen, S., & Vogt, P. (Eds.). (2003). *Usability praktisch umsetzen – Handbuch für Software, Web, Mobile Devices und andere interaktive Produkte* [Usability into practice - Manual for software, web, mobile devices and other interactive products]. München: Karl Hanser Verlag. In German

Hinzen, A. (1993): *The influence of human factors on railway safety* (in German), PhD thesis, RWTH Aachen, 1993

Hoare, C. (1985). *Communicating Sequential Processes*. Prentice Hall.

Hoenicke, J., & Olderog, E.-R. (2002). Combining Specification Techniques for Processes Data and Time. In M. Butler, L. Petre, & K. Sere, *Integrated Formal Methods* (pp. 245-266). Springer Verlag.

Hollnagel, E. (1993). *Human reliability analysis: Context and control*. London: Academic Press.

Hollnagel, E. (2004). *Barriers and Accident Prevention*. Hamshire, England: Ashgate Publishing Limited.

Hollnagel, E. (2008). Risk + Barriers = Safety? *Safety Science*, *46*, 221–229. doi:10.1016/j.ssci.2007.06.028

Holtzblatt, K., Burns Wendell, J., & Wood, S. (2005). *Rapid Contextual Design – A How-To Guide to Key Techniques for User-Centered Design*. San Francisco, California: Morgan Kaufmann Publishers.

Hong, X., & Dugan, J. B. (2004), Combining dynamic fault trees and event trees for probabilistic risk assessment, in *Reliability and Maintainability, 2004 Annual Symposium - RAMS*, (pp. 214- 219.)

Hood, J. N., Olivas, T., Slocter, C. B., Howard, B., & Albright, D. P. (2003), Vulnerability Assessment Through Integrated Transportation Analysis, in *Journal of the Transportation Research Board, 1822*, 18–23.

Huang, Y. C. (2003). Condition assessment of power transformers using genetic-based neural networks. *IEE Proceedings. Science Measurement and Technology*, *150*(1), 19–24. doi:10.1049/ip-smt:20020638

Huber, M., & King, S. (2002). Towards an integrated model checker for railway signalling data. InEriksson, L.-H. & Lindsay, P. (Eds.), *Proc. on Formal Methods Europe* (FME'2002), *volume2391*, (pp. 204-223.) Springer-Verlag.

Idani, A., Ossami, D.-D., & Boulanger, J.-L. (2007). Commandments of UML for Safety. In *International Conference on Software Engineering Advances, ICSEA* (p. 58). IEEE.

Idirin, M., Aizpurua, X., Villaro, A., Legarda, J., & Melendez, J. (2011, March). Implementation Details and Safety Analysis of a Microcontroller-based SIL-4 Software Voter. *IEEE Transactions on Industrial Electronics*, *58*(3). doi:10.1109/TIE.2010.2062471

IEC EN 61508 parts 1-7: 2003. (2003) *Functional safety of electrical/ electronic/ programmable electronic safety-related systems*.

IEC. (2010). *Functional safety of electrical/electronic/ programmable electronic safety-related systems (IEC 61508).*

IEEE. (1987). *IEEE Standard for a Versatile Backplane Bus: VMEbus*. Washington, DC: IEEE Press.

Ikeda, M., Suzuki, M., & Yoshida, K. (2005) Application of Jet Ejection to control Contact Force of Pantographs for High Speed Trains *Proc. of the 6th Symposium on Smart Control of Turbulence*, Tokio 06-09-2005

Images and technical documentation available at the official site of Railway Technical Research Institute http://www.rtri.or.jp/rtri/facility2_E.html

International Electrotechnical Commission. IEC (2011). Functional Safety and IEC 61508. Retrieved March 25, 2011, http://www.iec.ch/functionalsafety/

International Monetary Fund (IMF). (2010). World Economic Outlook. *Database*, *2010*, •••. retrieved from http://www.imf.org/external/pubs/ft/weo/2010/02/weodata/index.aspx

International Union of Railways (IUC). (2010) *Online Statistics* retrieved from http://www.uic.org/spip.php?article593

ISO (2010) DIS 26262: *Road vehicles – Functional safety*

ISO 13407 (1999)*Human-centered design processes for interactive systems*.

ISO 9241-110 (2006)*Ergonomics of human-system interaction - Part 110: Dialogue principles*.

JavaPathFinder. http://javapathfinder.sourceforge.net/

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer.

Jinfu, C., Yansheng, L., & Xiaodong, X. (2007) *Testing Approach of Component Security Based on Fault Injection*. Paper presented at the 2007 International Conference on Computational Intelligence and Security.

Jo, H., Hwang, J., & Yoon, Y. (2008): Applying formal method to train distance control system by combining ZED and Statechart, Proc. *of International Conference on Control Automation and Systems,* (pp. 896 – 900) 14-17 Oct. 2008.

Johnston, W., Winter, K., van den Berg, L., Strooper, P. A., & Robinson, P. (2006). Model-based variable and transition orderings for efficient symbolic model checking. In Misra, J., Nipkow, T. & Sekerinski, E. (Eds.), *Proc. of 14th Int. Symposium on Formal Methods* (FM 2006), *volume 4085 of LNCS*,(pp. 524-540.) Springer-Verlag.

Jordan, P. (2006, 17 February 2006). *A Trial of a Low Cost Level Crossing Warning Device.* Paper presented at the IRSE Annual Meeting.

Jordan, P. W. (2001). *An Introduction to Usability*. Philadelphia, PA: Taylor & Francis.

Kalker, J. J. (1991). Wheel-rail rolling contact theory. *Wear*, *144*(1-2), 243–261. doi:10.1016/0043-1648(91)90018-P

Kamhi, G., & Fix, L. (1998). Adaptive variable reordering for symbolic model checking. In *Proc. of IEEE/ACM Int. Conference on Computer-aided design* (ICCAD'98), (pp. 359-365.) ACM Press. Lewis, G., Comella-Dorda, S., Gluch, D., Hudak, J. & Weinstock, C. (2001). *Model-based verification: Analysis guidelines. Technical Report CMU/SEI-2001-TN-028*, Carnegie Mellon Software Engineering Institute.

Kaplan, S. (1992), Expert information versus expert opinions: another approach to the problem of eliciting/combining/using expert judgment in PRA, in Journal *of Reliability Engineering & System Safety, Elsevier, 35*, 61-72

Kaplan, S., & Garrick, G. J. (1981). On the quantitative definition of risk. In *Risk Analysis* (pp. 11–27). Wiley.

Kassakian, J. G., Schlecht, M. F., & Verghse, G. C. (1991). *Principles of power electronics*. Reading, MA: Addison-Wesley.

Kastner, C., Thum, T., Saake, G., Feigenspan, J., Leich, T., Wielgorz, F., & Apel, S. (2009). Featureide: A tool framework for feature-oriented software development. In *Proc. IEEE 31st Int. Conf. Software Engineering ICSE 2009*, pages 611–614.

Kelly, S., & Tolvanenm, J.-P. (2008). *Domain-Specific Modeling*. Hoboken, New Jersey: John Wiley & Sons Inc.doi:10.1002/9780470249260

Kent, S. (2002). Model driven engineering. In *Proceedings of the Third International Conference on Integrated Formal Methods, IFM '02*. 286-298, London, UK: Springer-Verlag.

Kia, S. H., Bartolini, F., Mpanda-Mabwe, A., & Ceschi, R. (2010), Pantograph-catenary interaction model comparison, *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, (pp.1584-1589), Washington, DC: IEEE Press.

Kim, K. I., Jung, K., & Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, *9*(2), 40–42. doi:10.1109/97.991133

Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic*. Prentice Hall PTR.

Knapp, A. (2004). *Semantics of UML State Machines.* München: Ludwig-Maximilians-Universität, Technical Report 0408.

Knapp, A., Merz, S., & Rauh, C. (2002). Model Checking Timed UML State Machines and Collaborations. In W. Damm, & E. Olderog, *Proc. 7th Int. Symp. Formal Techniques in Real-Time and Fault Tolerant Systems* (pp. 395-416). Berlin: Springer Verlag.

Kohl, R. J. (1999). Establishing guidelines for suitability of cots for a mission critical application. In *Annual International Computer Software and Applications Conference* (pp. 98-99). Phoenix, AZ, USA: IEEE Computer Society Publications.

Kranitis, N., Xenoulis, G., Paschalis, A., Gizopoulos, D., & Zorian, Y. (2003). Application and analysis of rt-level software-based self-testing for embedded processor cores. In *International Test Conference* (pp. 431–440). Charlotte, NC, USA: IEEE Computer Society Publications.

Kranitis, N., Paschalis, A., Gizopoulos, D., & Xenoulis, G. (2005). Software-based self-testing of embedded processors. *IEEE Transactions on Computers*, *54*(4), 461–475. doi:10.1109/TC.2005.68

Krstic, A., Lai, W.-C., & Cheng, K.-T. & Chen &. & Dey S. (2002). Embedded software-based self-test for programmable core-based designs. *IEEE Design & Test of Computers*, *19*(4), 18–27. doi:10.1109/MDT.2002.1018130

Kuniavsky, M. (2003). *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco, CA: Morgan Kaufmann Publishers.

Kurita, T,. Hara, M., Yamada, H., Wakabayashi, Y., Mizushima, F., Satoh, H., & Shikama, T. (2010) Reduction of Pantograph Noise of High-Speed Trains, *Journal of Mechanical Systems for Transportation and Logistics* [23], *3*(1) Special issue on STECH'09.63-74

Kwiatkowska, M., Norman, G., & Parker, D. (2011). PRISM 4.0: Verification of Probabilistic Real-time Systems (to appear). In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*. Springer Verlag.

Lambert, J. H., & Farrington, M. W. (2007), Cost-benefit functions for the allocation of security sensors for air contaminants, in *Journal of Reliability Engineering & System Safety, 92*, 930-946.

Latvala, T., & Heljanko, K. (2000). Coping with Strong Fairness. In *Fundamenta Informaticae* (pp. 175–193). IOS Press.

Lecomte, T. (2008). *LNCS 5014: FM 2008* (pp. 430-434), Turku, Finland. Berlin, Germany: Springer.

Lee, W. S., Grosh, D. L., Tillman, F. A., & Lie, C. H. (1985), Fault Tree Analysis, Methods, and Applicationn: A Review, in *IEEE Transactions on Reliability, 34*(3),194-203.

Lehrasab, N., Dassanayake, H. P. B., Roberts, C., Fararooy, S., & Goodman, C. J. (2002). Industrial fault diagnosis: Pneumatic train door case study. *Proceedings of the IMechE, Part F: Rail and Rapid Transit, 215*(11), 27-46.

Leinenbach, D., & Santen, T. (2009). Verifying the Microsoft Hyper-V Hypervisor with VCC. In Cavalcanti, A., & Dams, D. R. (eds.), *Proceedings of the 2nd World Congress on Formal Methods*, (pp. 806-809), Berlin Heidelberg: Springer-Verlag.

Leitsch, R. (1995). *Reliability Analysis for Engineers: An Introduction*. Oxford Science Publications.

Leuschel, M., Falampin, J., Fritz, F., & Plagge, D. (2009) Automated Property Verification for Large Scale B Models. *LNCS 5850: FM 2009* (pp. 708-723). Eindhoven, The Netherlands. Berlin, Germany: Springer.

Leveson, N. (1995). *Safeware: System Safety and Computers*. Addison-Wesley Publishing.

Leveson, N. (2002). *A new approach to system safety engineering*. Aeronautics and Astronautics Massachusetts Institute of Technology.

Lichtenberg, G., & Lunze, J. (1997). Observation of qualitative states by means of a qualitative model. *International Journal of Control*, *66*(6), 885–903. doi:10.1080/002071797224441

Li, D., Pedrycz, W., & Pizzi, N. J. (2005). Fuzzy wavelet packet based feature extraction method and its application to biomedical signal classification. *IEEE Transactions on Bio-Medical Engineering*, *52*(6), 1132–1139. doi:10.1109/TBME.2005.848377

Lin, C. E., Ling, J. M., & Huang, C. L. (2003). An expert system for transformer fault diagnosis using dissolved gas analysis. *IEEE Transactions on Power Delivery*, *8*(1), 231–238. doi:10.1109/61.180341

Lindgaard, G., & Chattratichart, J. (2007). Usability testing: what have we overlooked? In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. (pp. 1415-1424). New York, NY: ACM.

Liu, W., Luo, G., Zhao, N., & Dou, M. (2009, September). *Design and HIL simulation of proportional compression salient-pole permanent magnet synchronous motor for electrical traction vehicle*. Paper presented at the 5th IEEE Vehicle Power and Propulsion Conference, Dearborn, Michigan.

Löding, H., & Peleska, J. (2010). Timed Moore automata: test data generation and model checking. In *Proceedings of the Third International Conference on Software Testing, Verification and Validation ICST,* (pp. 449-458). DOI http://doi.ieeecomputersociety.org/10.1109/ICST.2010.60

Luenberger, D. (1989). *Linear and nonlinear programming*. Addison-Wesley Publishing.

Lunze, J. (1992). Qualitative modelling of continuous-variable systems by means of non-deterministic automata. *Intelligent Systems Engineering*, *1*(1), 22–30. doi:10.1049/ise.1992.0003

Lunze, J. (2000). Diagnosis of quantized systems based on a timed discrete-event model. *IEEE Transactions on Systems, Man, and Cybernetics*, *30*(3).

Lunze, J., Nixdorf, B., & Schröder, J. (1999). Deterministic discrete-event representations of linear continuous-variable systems. *Automatica*, *35*, 395–406. doi:10.1016/S0005-1098(98)00176-9

MAAB. (2007). *Control Algorithm Modelling Guidelines Using Matlab, Simulink and Stateflow*, Version 2.0. Retrieved August 7, 2009, from http://www.mathworks.com/industries/auto/maab.html

Malvezzi, M., Allotta, B., & Pugi, L. (2008). Feasibility of degraded adhesion tests in a locomotive roller rig. *Institute of Mechanical Engineers. Part F: Journal of Rail and Rapid Transit*, *222*(1), 27–43. doi:10.1243/09544097JRRT108

Manna, Z., & Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag. doi:10.1007/978-1-4612-0931-7

Marcano, R., Colin, S., & Mariano, G. (2004). A Formal Framework for UML Modelling with Timed Constraints: Application to Railway Control Systems. *LNCS 3297: SVERTS: Specification and Validation of UML models for Real Time and Embedded Systems*, (pp.33-38) Lisbonne, Portugal. Berlin, Germany: Springer.

Martins, M., & Rosa, A. C. A. (2000). *A Fault Injection Approach Based on Reflective Programming*. Paper presented at the 2000 IEEE/IFIP International Conference on Dependable Systems and Networks.

Massat, J. P., Bobillot, A., & Laine, J. P. (2006) Robust Methods for Detecting Defects in Overhead Contact Line Based on Simulation Results, *Proceedings of III European Conference on Computational Mechanics 2006,* The Netherlands: Springer - 978-1-4020-5370-2

Matar, M., & Iravani, R. (2010). FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients. *IEEE Transactions on Power Delivery*, *25*(2), 852–860. doi:10.1109/TPWRD.2009.2033603

McMillan, K. (1993). *Symbolic Model Checking*. Kluwer Academic Publishers. doi:10.1007/978-1-4615-3190-6

Medjoudj, M., & Yim, P. (2007). Extraction of Critical Scenarios in a Railway Level Crossing Control System. *International Journal of Computers, Communications & Control*, ▪▪▪, 252–268.

Meli, E., Malvezzi, M., Papini, S., Pugi, L., Rinchi, M., & Rindi, A. (2008). A railway vehicle multibody model for real-time applications. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, *46*(12), 1083–1105. doi:10.1080/00423110701790756

Metayer, C., & Clabaut, M. (2008). DIR 41 Case Study. *LNCS 5238: Abstract State Machines, B and Z, First International Conference,* (p. 357), London, UK. Berlin, Germany: Springer.

Mewes, K. (2010). *Domain-specific Modelling of Railway Control Systems with Integrated Verification and Validation*. München: Verlag Dr. Hut.

Midya, S., Bormann, D., Schütte, T., & Thottappillil, R. (2009). Pantograph arcing in electrified railways - mechanism and influence of various parameters - Part I: with DC traction power supply. *IEEE Transactions on Power Delivery*, *24*(4), 1931–1939. doi:10.1109/TPWRD.2009.2021035

Milius, B. (2010). *Construction of a semi-quantitative risk graph* (in German), PhD thesis, Technical University of Brunswick

Miller, S. P., Whalen, M. W., & Cofer, D. (2010). Software model checking takes off. *Communications of the ACM*, *53*(2), 58–64. doi:10.1145/1646353.1646372

Mohan, N., Undeland, T. M., & Robbins, W. P. (Eds.). (2003). *Power Electronics: Converters, Applications, and Design*. John Wiley and Sons.

Moon, I., Hachtel, G. D., & Somenzi, F. (2000). Border-block triangular form and conjunction schedule in image computation. *In Proc. of Formal Methods in Computer-Aided Design* (FMCAD 2000), *volume 1954 of LNCS*,(pp. 73-90), Springer-Verlag.

Mori, K., Kasashima, N., Yoshioka, T., & Ueno, Y. (1996). Prediction of spalling on a ball bearing by applying the discrete wavelet trasform to vibration signals. *Wear*, *195*, 162–168. doi:10.1016/0043-1648(95)06817-1

Mpanda, A. (2009)Real-Time Test Rig & HIL Simulation Platform for Testing Pantograph-Catenary Interaction, *Proceedings of the International Seminar-Workshop on Power Transmission in High Speed Railway Systems*, Amiens, France

Murata, T. (1989), Petri Nets: Properties, Analysis and Applications, in *Proceedings IEEE, 77*. 541-580.

Murray-Tuite, P. M. (2007), Transportation Network Risk Profile for an Origin-Destination Pair: Security Measures, Terrorism, and Target and Attack Method Substitution, in *Proceedings of 87th TRB Annual Meeting.* TRB

National Transportation Safety Board (NTSB) (1993) *Railroad Accident Report (RAR-94-01): Derailment of Amtrak Train No. 2 on the CSXT Big Bayou Canot Bridge Near Mobile, Alabama September 22, 1993.*

National Transportation Safety Board (NTSB) (2008) *NTSB Most Wanted List Transportation Safety Improvements, 2008-2009*

National Transportation Safety Board (NTSB). (2003) *Railroad Accident Report (RAR-03-04): Collision of Burlington Northern Santa Fe Freight Train With Metrolink Passenger Train, Placentia, California, April 23, 2002.*

National Transportation Safety Board (NTSB). (2005) *Railroad Accident Report (RAR-05-04): Collision of Norfolk Southern Freight Train 192 With Standing Norfolk Southern Local Train P22 With Subsequent Hazardous Materials Release at Graniteville, South Carolina, January 6, 2005*

National Transportation Safety Board (NTSB). (2010) *Railroad Accident Report (RAR-01-01): Collision of Metrolink Train 111 with Union Pacific Train LOF65–12 Chatsworth, California, September 12, 2008*

Nemhauser, G., Rinnooy Kan, A., & Todd, M. (1989). *Optimization*. ElsevierScience Publishers B.V.

NERA. (2000). *Safety Regulations and Standards for European Railways*. NERA.

Nielsen, J. (1993). *Usability Engineering*. San Diego, California: Academic Press.

Niska, S. (2008) *Measurements and analysis of electromagnetic interferences in the Swedish railway systems*. Doctoral thesis Luleå tekniska universitet

Norman, D. A. (2002). *The design of everyday things*. New York: Basic Book.

NuSMV. (n.d.), Retrieved from: http://nusmv.fbk.eu

Obama, B. (2008). *Strengthening Americans Transportation Infrastructure*. Retrieved from.http://www.barackobama.com/pdf/issues/FactSheetTransportation.pdf

Object Management Group. (2001). *Model Driven Architecture (MDA). Technical report*. Framingham, MA: Object Management Group.

Object Management Group. (2005). *Unified Modeling Language: Superstructure (version 2.0). Technical report*. Framingham, MA: Object Management Group.

Object Management Group. OMG (2010). *OMG Unified Modeling Language (OMG UML), Infrastructure, V2.3.* Retrieved April 1, 2011, from http://www.omg.org/spec/UML/2.3/

Object Management Group. OMG (2010). *OMG Unified Modeling Language (OMG UML), Superstructure, V2.3*. Retrieved April 1, 2011, from http://www.omg.org/spec/UML/2.3/

Office of Rail Regulation. HSE (2011). *Railway Safety Principles and Guidance, Part 2, Section D, Guidance on Signaling*. Retrieved March 25, 2011, http://www.rail-reg.gov.uk/upload/pdf/rspg-2d-signlng.pdf

OMG. (2010). *Unified Modeling Language Specification formal/2010-05-03.* Version 2.3.

Ortmeier, F., Schellhorn, G., & Reif, W. (2004). Safety Optimization of a Radio-Based Railroad Crossing. In E. Schnieder, & G. Tarnai, *Formal Methods for Automation and Safety in Railway and Automotive Systems.* Braunschweig.

Özyurt, I. B., Hall, L. O., & Sunol, A. K. (1999). SQFDiag: Semiquantitative model-based fault monitoring and diagnosis via episodic fuzzy rules. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans*, *29*(3), 294–306. doi:10.1109/3468.759283

Parka, T. J., Han, C. S., & Jan, J. H. (2003). Dynamic sensitivity analysis for the pantograph of a high-speed rail vehicle. *Journal of Sound and Vibration*, *266*, 235–26. doi:10.1016/S0022-460X(02)01280-4

Parvathala, P., Maneparambil, K., & Lindsay, W. (2002). FRITS - a microprocessor functional BIST method. In IEEE *International Test Conference* (pp. 590–598). Baltimore, MD, USA: IEEE Computer Society Publications.

Paschalis, A., & Gizopoulos, D. (2004). Effective software-based self-test strategies for on-line periodic testing of embedded processors. In *Design, Automation and Test in Europe Conference and Exhibition* (pp. 578–583). Paris, France: IEEE Computer Society Publications.

Paschalis, A., Gizopoulos, D., Kranitis, N., Psarakis, M., & Zorian, Y. (2001). Deterministic software-based self-testing of embedded processor cores. In *Design, Automation and Test in Europe, 2001. Conference and Exhibition 2001. Proceedings* (pp. 92-96)

Pasetti, A. (2002). *Software Frameworks and Embedded Control Systems, volume 2231 of Lecture Notes in Computer Science*. Springer.

Patton, R., & Chen, J. (1992). A robustness study of model-based fault diagnosis for jet engine systems. *Proceedings of 1st IEEE Conference on Control Applications,* (pp. 871-876).

PD CLC/TR 50507(2007) *Railway applications. Interference limits of existing track circuits used on European railways*. Dolecek,R. & Hlava, K. (2007): Transient Effects at Power-Supply System of the Czech Railways from EMC Viewpoint.. *RADIOENGINEERING16*(1)

Pearce, J. (2000) What's All This Eb/No Stuff, Anyway?, *Spread Spectrum Scene Online 7*(1)

Pedicini, C., Vasca, F., Iannelli, L., & Jonsson, U. (2011, December). *An overview on averaging for pulse-modulated switched systems*. Paper accepted for presentation at the 50th IEEE Conference on Decision and Control, Orlando, Florida, USA.

Peleska, J., & Haxthausen, A. E. (2007). Object code verification for safety-critical railway control systems. *Formal methods for automation and safety in the railway and automotive systems (FORMS/FORMAT 2007).* Braunschweig, Germany: GZVB e.V.

Peterman, D. R. (2006). Overview of Issues. In *CRS Report for Congress*. Passenger Rail Security.

Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*. NJ, USA: Prentice Hall PTR.

Piccolo, A., Senesi, F., Galdi, V., & Malangone, R. (2009): Use of Formal Language to represent the ERTMS/ETCS system requirements Specification of the Interconnection L0/L2 - *International Conference on Model and Technologies for intelligent transportation systems*, Università La Sapienza in Roma 22-23 June 2009.

Pittner, S., & Kamarthi, S. V. (1999). Feature extraction from wavelet coefficients for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(1), 83–88. doi:10.1109/34.745739

Platzer, A., & Quesel, J.-D. (2009). European Train Control System: A Case Study in Formal Verification. In K. Breitman & Cavalcanti, A. (eds.), *11th Conference on Formal Engineering Methods ICFEM 2009*: LNCS 5885 (pp. 246-265). Berlin, Heidelberg: Springer.

Pnueli, A., Shtrichman, O., & Siegel, M. (1998). The Code Validation Tool CVT: Automatic verification of a compilation process. *International Journal on Software Tools for Technology Transfer*, *2*(1), 192–201.

Polet, P., Vanderhaegen, F., & Wieringa, P. A. (2002). Theory of safety-related violations of system barriers. *Cognition Technology and Work*, *4*, 171–179. doi:10.1007/s101110200016

Prover Technologies. (n.d.), Retrieved from: http://www.prover.com/

Prover Technology, A. B. (2007). *Prover iLock*. Company Whitepaper.

Psarakis, M., Gizopoulos, D., Sanchez, E., & Reorda, M. S. (2010). Microprocessor Software-Based Self-Testing. *IEEE Design & Test of Computers*, *27*(3), 4–19. doi:10.1109/MDT.2010.5

Public Transport Safety Victoria. (2006). *Appendix G: Risk management and 'so far as is reasonably practicable' (SFARP)*. Accreditation Guideline.

Pugi, L., & Rinchi, M. (2002) test Rig for Train Brakes, *3rd AIMETA International Tribology Conference*, AITC, Salerno - Italy 18 - 20 September 2002.

Pugi, L., Malvezzi, M., & Tarasconi, A. Palazzolo, A., Cocci, G, & Violani, M. (2006)Simulation of WSP Systems on MI-6 Test Rig, *Vehicle system and dynamics* (Taylor and Francis) *44*, 843-852, ISBN 978-0-415-43616-8

Pugi, L., Malvezzi, M., Tarasconi, A., Palazzolo, A., Cocci, G., & Violani, M. (2005b, August). *HIL simulation of WSP systems on MI-6 test rig*. Paper presented at 19th Symposium of the International Association for Vehicle System Dynamics, Milan, Italy.

Pugi, L., Ridolfi, A., Allotta, B., Malvezzi, M., Vettori, G., Cuppini, F., & Salotti, F. (2011)*A 3D Simulation Model of Train Dynamics for Testing Odometry Algorithms,* Proceedings of WCRR 2011 (World Congress on Railway Research), Lille,France 22-26 May 2011

Pugi, L., Rinchi, M., Malvezzi, M., & Cocci, G. (2005a, July). *A multipurpose platform for HIL testing of safe relevant railway subsystem*. Paper presented at the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA.

Queensland Rail Signal and Operational Systems. (1998). *Signalling Principles - Brisbane SuburbanArea. Technical Report S0414,* Queensland Rail Technical Services Group.

RAILCOM (2008): *Granted European Project: Electromagnetic compatibility between rolling stock and rail-infrastructure encouraging European interoperability*. Cordis FP6

Railroad Safety Advisory Committee (RSAC) (1999) *Implementation of Positive Train Control Systems*

Railway Applications (2002) *Current Collection Systems—Validation of Simulation of the Dynamic Interaction Between Pantograph and Overhead Contact Line, European Standard EN 50318*, Jul. 2002.

Railway Industry Safety and Standards Board. (2009). *Level Crossing Stocktake*. Retrieved from http://www.rissb.com.au/userfiles/file/RLX/2009 Level Crossing Stocktake Report.pdf

Raina, R., & Molyneaux, R. (1998). Random self-test method applications on PowerPC microprocessor caches microprocessor caches. In *8th Great Lakes Symposium on VLSI* (pp. 222–229). Lafayette, LA, USA: IEEE Computer Society Publications.

Raj, P. K., & Pritchard, E. W. (2000) Hazardous Materials Transportation on US Railroads, in *Transportation Research Record, 1707*.

Ramkumar, K. B., Philips, P., Presig, H. A., Ho, W. K., & Lim, K. W. (1998). Structured fault-detection and diagnosis using finite state-automaton. *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, (pp. 1667-1672).

Raskin, J. (2000). *The humane interface – New directions for designing interactive systems*. Crawfordsville, IN: ACM.

Ravi, V., Redd, P. J., & Zimmermann, H. (2000), Fuzzy global optimisation of complex system reliability, in *IEEE Transactions on Fuzzy Systems, 8*(3), 241–248.

Reason, J. (1990). *Human Error*. New York, USA: Cambridge University Press.

Reifer, D. (1979). Software failure modes and effects analysis. *IEEE Transactions on Reliability*, ▪▪▪, 147–249.

Ren, W., Sloderbeck, M., Steurer, M., Dinavahi, V., Noda, T., & Filizadeh, S. (2011). Interfacing issues in real-time digital simulators. *IEEE Transactions on Power Delivery*, *26*(2), 1221–1229. doi:10.1109/TPWRD.2010.2072792

RFI-DTC/DNS/EE.ST TE 74 D (February, 2008), *Prove da eseguire per caratterizzazione di un pantografo a 3kV CC* [Tests to be performed for the characterization of a pantograph 3kV DC], Sciavicco, L., and Siciliano, B. (1996) *Modelling and Control of Robot Manipulators*. New York: McGraw-Hill.

RIAC. (2010) *RIAC's Relaiability Prediction Methodology 217 plus™*. Retrieved from: http://www.theriac.org/riacapps/search/?mode=displayresult&id=351

Roberts, C., Dassanayake, H. P. B., Lehrasab, N., & Goodman, C. J. (2002). Distributed quantitative and qualitative fault diagnosis: Railway junction case study. *Control Engineering Practice*, *10*(4), 419–429. doi:10.1016/S0967-0661(01)00159-9

Robinson, N., Barney, D., Kearney, P., Nikandros, G., & Tombs, D. (2001). Automatic generation and verification of design specification. In *Proc. of Int. Symp. of the International Council On Systems Engineering* (INCOSE).

Roop, S. S., Roco, C. E., Olson, L. E., & Zimmer, R. A. (2005). *An Analysis of Low-Cost Active Warning Devices for Highway-Rail Grade Crossings* (Institute, T. T., Trans.). Texas: Texas A&M University.

Rosson, M. B., & Carrol, J. M. (2002). *Usability Engineering – Scenario-Based Development of Human-Computer Interaction.* San Francisco, California: Morgan Kaufmann Publishers.

Rowshan, S., Sauntry, W. C., Wood, T. M., Churchill, B., & Levine, S. R. (2005). *Transportation Research Record, 1938*. Reducing Security Risk for Transportation Management Center.

RSSB. (2004). *Rail-Specific Human Reliability Assessment Technique for Driving Tasks, T270 Final Report*, Rail Safety and Standard Board,Retrieved from: www.rssb.co.uk.

RTCA. (1992). *Software Considerations in Airborne Systems and Equipment Certification (DO-178B)*.

Salter, P. (2008). National Guideline for the Meaning of Duty to Ensure Safety So Far As Is Reasonably Practicable *National Railway Safety Guideline*: National Transportation Commission. Retrieved from http://www.rsrp.asn.au/files/legislation/42_7.pdf

Sarodnick, F., & Brau, H. (2006). *Methoden der Usability-Evaluation - Wissenschaftliche Grundlagen und praktische Anwendungen* [Methods of usability evaluation - scientific fundamentals and practical applications]. Bern: Hans Huber. In German

Sauvage, S., & Bouali, A. (2006) Development Approaches in Software Development. *Proceedings of ERTS*, Toulouse, France.

SCADE. (n.d.), Retrieved from: http://www.esterel-technologies.com/

Schiller, F., Schröder, J., & Lunze, J. (2001). Diagnosis of transient faults in quantised systems. *Engineering Applications of Artificial Intelligence*, *14*, 519–536. doi:10.1016/S0952-1976(01)00020-3

Schlich, B. (2010). Model Checking of Software for Microcontrollers. *ACM Transactions in Embedded Computing Systems*, *9*(4), 1–27. doi:10.1145/1721695.1721702

Schmidt, D. C. (2006). Model-Driven Engineering. *IEEE Computer*, *39*(2), 25–31. doi:10.1109/MC.2006.58

Senesi F. & Malangone R.(2010): Ram analysis of the Radio Block Center System for Italian ERTMS lines, *Ingegneria Ferrviaria [Railway Engineering]*, April 2010

Senesi F.(2009b): Sistemi di protezione e controllo della marcia dei treni (ERTMS, SCMT, SSC): applicazioni e sviluppo per la rete ferroviaria nazionale italiana [Train running protection control systems (ETCS, SCMT, SSC): application and development for the italian railway system], *Ingegneria Ferroviaria [Railway Engineering]*, April 2009

Senesi F., Bonafè G., Geraci S., Frandi M., Filippini N. & Malangone R.(2008): Getting ERTMS into service quicker ETR 500 test train, *Railway Gazette International*, December 2008.

Senesi F., Malangone R., Piccolo A., & Galdi V.(2006a): Utilizzo di linguaggi formali per la analisi e la valutazione delle specifiche di test del sistema ERTMS della rete italiana ad Alta Velocità, [Using formal languages for analysis and evaluation of the test specifications of the ERTMS on the Italian network to High Speed *]Ingegneria Ferroviaria*, [Railway Engineering] December 2006.

Senesi F., Malangone R., Piccolo A., & Galdi V.(2006b): Utilizzo di linguaggi formali per la analisi e la valutazione delle specifiche di test del sistema ERTMS della rete italiana ad Alta Velocità, [Using formal languages for analysis and evaluation of the test specifications of the ERTMS on the Italian network to High Speed] *Ingegneria Ferroviaria* [*Railway Engineering*], December 2006.

Senesi F., Malangone R., Rossi C., & Torassa M.(2007c): Le Apparecchiature del Sistema Controllo Marcia Treni [The Equipment Control System Run Trains], *Tecnica Professionale* [Professional Techniques], October 2007

Senesi, F., & Malangone, R. (2007a): Formal method analysis and evaluation of ERTMS-TEST specification for the Italian high speed railway, *Proc. of Formal 6th International Symposium on Methods for automation and safety in railway and automotive systems - FORMS/ FORMAT 2007*, 25-26 January 2007.

Senesi, F., & Marzilli, E. (2007b): *European Train Control System - Development and Implementation in Italy* Retrieved from: http://www.etcsbook.com.

Senesi, F., Filippini, N. & Malangone, R. (2009a) RFI – SAFETY & SIGNALLING ON FLORENCE-BOLOGNA, *Eurailmag 03*

Senesi, F., Malangone, R., & Petaccia, G. (2006c): Utilizzo della distanza obiettivo come seconda catena di appuntamento nella Logica SCMT, Ingegneria Ferroviaria, ed. settembre 2006.

Senesi, M. (2009c): Il progetto GRIDES per la disponibilità della rete GSM-R, [The project GRIDES for the availability of GSM-R] *Tecnica Professionale* [Profession Techniques], September 2009

Shen, J., & Abraham, J. A. (1998). Native mode functional test generation for processors with applications to self-test and design validation. In *International Test Conference* (pp. 990–999). Washington, DC, USA: IEEE Computer Society Publications.

Silmon, J. A. (2009). *Quantification of benefit available from switch and crossing monitoring.* INNOTRACK Research Project Deliverable. Retrieved from www.innotrack.eu

Silmon, J. A., & Roberts, C. (2006). A systems approach to fault detection and diagnosis for condition-based maintenance. *Proceedings of the 1st IET International Conference on Railway Condition Monitoring*, 2006.

Simpson, A., Woodcock, J., & Davies, J. (1997). The mechanical verification of solid state interlocking geographic data. In Groves, L. & Reeves, S. (Eds.), *Proc. of Formal Methods Pacific* (FMP'97), Discrete Mathematics and Theoretical Computer Science Series, pp. 223-243. Springer-Verlag.

SIMULINK. (n.d.), Retrieved from: http://www.mathworks.com/products/simulink/

Skarin, D., Barbosa, R., & Karlsson, J. (2010). *GOOFI-2: A Tool for Experimental Dependability Assessment.* Paper presented at the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks.

Sklet, S. (2006). Safety barriers: Definition, classification, and performance. *Journal of Loss Prevention in the Process Industries*, *19*, 494–506. doi:10.1016/j.jlp.2005.12.004

Slimen, N., & Deniau, V. (2008): On Board Measurements of the Railway's Electromagnetic Noise with Moving Train. In Slimen, N., Deniau, V., Baranowski, S., Rioult, J., Dubalen, N., & Démoulin, B. (eds.) *Consortium Railcom. Proceedings, 18th Int. Zurich Symposium on EMC*, Munich

Slotine, J. J., & Li, W. (1991). *Applied Non Linear Control*. Prentice Hall.

SMV. (n.d.), Retrieved from: http://www.cs.cmu.edu/~modelcheck/smv.html

Sophian, A., Tian, G. Y., Taylor, D., & Rudlin, J. (2003). A feature extraction technique based on Principal Component Analysis for pulsed eddy current NDT. *NDT & E International*, *36*, 37–41. doi:10.1016/S0963-8695(02)00069-5

Sosnowski, J. (2006). Software-based self-testing of microprocessors. *Journal of Systems Architecture*, *52*, 257–271. doi:10.1016/j.sysarc.2005.05.004

SPIN. (n.d.), Retrieved from: http://spinroot.com/spin/whatispin.html

Srinivas, M., & Patnaik, L. M. (1994), Genetic algorithms: a survey, in *Computer, 27*, 17-26.

Stallings, W. (2008). *Operating systems: internals and design principles*. Upper Saddle River, NJ: Prentice Hall.

Stamenkovic, B. B., & Dersin, P. (2009). Availability Assessment of ALSTOM's safety-relevant trainborne odometry sub-system. In Martorell, S. (Ed.), *Safety, Reliability and Risk Analysis: Theory, Methods and Applications 2009*. London, UK: Taylor & Francis Group.

Standards Australia. (1999). *AS 61508.1-1999 Functional safety of electrical/electronic/programmable electronic safety-related systems: Part 1: General requirements*. Homebush, NSW, Australia: Standards Australia.

Standards Australia. (2007). *Manual of uniform traffic control devices Part 7: Railway crossings.*

Stanley, P. (2011). *ETCS for Engineers*. Hamburg, Germany: EurailPress.

Std, I. E. E. E. 610.12-1990 (1990) *IEEE Standard Glossary of Software Engineering Terminology* Washington, DC: IEEE Press

Steinberg, D., Budinsky, F., Paternostro, M., & Merks, E. (2008). *EMF: Eclipse Modeling Framework*. Addison-Wesley Professional.

Storey, N. (1996). *Safety critical computer systems*. Upper Saddle River, NJ: Prentice Hall.

Sträter, O. (1997) *Evaluation of Human Reliability on the Basis of Operational Experience*, PhD thesis, Technical University of Munich

Sträter, O. (2005). *Cognition and Safety – An Integrated Approach to System Design and Assessment*. Hampshire, England: Ashgate Publishing Limited.

Strauss, J. F. (1998). *The Burlington Northern An Operational Chronology 1970-1995, Friends of the Burlington Northern Railroad*. Wisconsin: West Bend.

SUBSET036. (2005). *FFFIS for Eurobalise SUBSET-036. (2.3.2)*. UNISIG.

SUBSET044. (2004). *FFFIS for Euroloop SUBSET-044 (2.3.0.)*. UNISIG.

Sveinsson, J. R., Ulfarsson, M. O., & Benediktsson, J. A. (2001). Cluster-based feature extraction and data fusion in the wavelet domain. *IEEE International Geoscience and Remote Sensing Symposium*, (pp. 867-869).

Tanenbaum, A. S. (2008). *Modern Operating Systems*. Upper Saddle River, NJ: Pearson.

Tellini, B., Schneider, M., Petri, A., & Ciolini, R. (2008) Measurements of EM Emission in Rail Launcher Operation. *I2MTC 2008 - IEEE International Instrumentation and Measurement Technology Conference* Washington, DC: IEEE Press

Terwiesch, P., Keller, T., & Scheiben, E. (1999). Rail vehicle control system integration testing using digital hardware-in-the-loop simulation. *IEEE Transactions on Control Systems Technology*, *7*(3), 352–362. doi:10.1109/87.761055

Thatte, S. M., & Abraham, J. A. (1980). Test generation for microprocessors. *IEEE Transactions on Computers*, *29*(6), 429–441. doi:10.1109/TC.1980.1675602

Toni, P., Malvezzi, M., Pugi, L., Rinchi, M., & Presciani, P. (2003)"Sviluppo e validazione di algoritmi di odometria per sistemi di controllo e monitoraggio ferroviari," [Development and validation of algorithms for odometry systems control and monitoring rail] *Ingegneria Ferroviaria* [Railway Engineering] 433/457 (in Italian).

Tooth, R., & Balmford, M. (2010). *Railway Level Crossing Incident Costing Model: Railway Industry Safety and Standards Board*. RISSB.

Travé-Massuyès, L., & Milne, R. (1997). Gas-turbine condition monitoring using qualitative model-based diagnostics. *IEEE Expert*, ▪▪▪, 22–31. doi:10.1109/64.590070

TREND. (2007): *Test of Rolling Stock Electromagnetic Compatibility for cross-Domain Interoperability*. Research Project funded by the European Community's Framework Programme FP7/2007–2013 under grant agreement nº 285259. Consortium: CEIT (E), CAF group (E), CEDEX (E), IFSTTAR (F), YORK EMC Services (UK), LTU (SE) and Trafikverket (SE).

Trenitalia S.p.A., (2000) SCMT, progetto dell'algoritmo per il calcolo della velocità istantanea del treno e dello spazio percorso, [SCMT, algorithmic project for the calculation of the instantaneous speed of the train and the distance traveled], *Unità Tecnologie Materiale Rotabile*, UTMR.DT.PS.31-10-2000

Trowitzsch, J., & Zimmermann, A. (2006). Using UML state machines and petri nets for the quantitative investigation of ETCS. In *Proceedings of the 1st international conference on performance evaluation methodolgies and tools* (pp. 34–es).

U.S. Department of Transportation (DOT) Bureau of Transportation Statistics. (2010) *National Transportation Statistics 2010* retrieved from http://www.bts.gov/publications/national_transportation_statistics

U.S. Government Printing Office (GPO) (2005) Standards for Development and use of Processor based Signal and train Control Systems, Final Rule, *Federal Register, 70*(43), U.S. Government Printing Office (GPO) (2008) *PUBLIC LAW 110–432—OCT. 16, 2008 FEDERAL RAIL SAFETY IMPROVEMENTS- Rail Safety Improvement Act of 2008*

U.S. Government Printing Office (GPO). (2010). Positive Train Control Systems Final Rule. *Federal Register*, *75*(10).

UIC 541-05 (2005) *Brakes - Specifications for the construction of various brake parts - Wheel Slide Protection device (WSP)* (2nd ed.), November 2005 – Translation List of recent publications1/06 (date of issue 1/02/2006) ISBN2-7461-0969-7

UIC 541-4 (May, 2007) *Brakes - Brakes with composition brake blocks - General conditions for certification of composite brake blocks* (3rd ed)UIC

UIC541-3 (2010) *Brakes - Disc brakes and disc brake linings, edition(7th ed.)* UIC

UNIFE. (2011) *ERTMS projects. Website*. Retrieved August 05, 2011from: http://www.ertms.com/2007v2/projects.html.

UNISIG (2007), Subset 085, *Test Specification for Eurobalise FFFIS*, Issue: 2.2.2.

UNISIG (2008). *ERTMS/ETCS - Baseline 3: System Requirements Specification*: Subset-026: Version 3.0.0, 23.12.2008.

UNISIG (2009), Subset 091, *Safety Requirements for the Technical Interoperability of ETCS in Levels 1 & 2*, Issue 2.5.0.

UNISIG (2009c), Subset 094, *Functional Requirements for an on board Reference Test Facility, Issue 2.0.2*.

UNISIG, (2007a) Subset036, *FFFIS for Eurobalise*, Issue 2.4.1.

UNISIG, (2009a) Subset076, *ERTMS/ETCS Class 1, test plan*, Issue: 2.3.1.

UNISIG, (2010) Subset026, *System requirement specification*, Issue 3.0.0.

UNISIG. (2002), *ERTMS/ETCS – Class 1 – System Requirement Specification, 026*,(2.2.2)

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation,026-1*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-2*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-3*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, -026-4*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-5*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-7*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2006). *ERTMS/ETCS – Class 1 System Requirements Speci□cation, 026-8*(2.3.0). Retrieved April 3, 2011, from http://www.era.europa.eu/Document-Register/Documents/SUBSET-026-SRS%20230.zip

UNISIG. (2008). ERTMS/ETCS - Class 1: Test Specification. *Subset*, *076*, 2008.

UNISIG. (2009). */ETCS– Class 1 Test Plan. 076-0*(2.3.1). Retrieved April, 9, 2011, from http://www.era.europa.eu/Document-Register/Pages/UNISIG%20SUBSET-076-0.aspx

UNISIG.(2009) *ERTMS/ETCS - Class 1: Functional Requirements for an on board Reference Test Facility*: Subset-094: Version 2.0.2, 05.02.2009.

US Congress, (2007), *S. 184, The Surface Transportation and the Railway Security Act of 2007*.

Ustun, V., Yapicioglu, H., Gupta, S., Ramesh, A., & Smith, J. S. (2005), A Conceptual Architecture for Static Features in Physical Security Simulation, in *Proceedings of the 2005 Winter Simulation Conference.*

Utamaphethai, N., Blanton, R., & Shen, J. (1999). Superscalar processor validation at the microarchitecture level. In *20th International Conference On VLSI Design* (pp. 300–305). Goa, India: IEEE Computer Society Publications.

Utting, M., & Legeard, B. (2006). *Practical Model-Based Testing: A Tools Approach*. Morgan-Kaufmann.

van den Berg, L., Strooper, P., & Johnston, W. (2007). An automated approach for the interpretation of counter-examples. *Electronic Notes in Theoretical Computer Science*, *174*(4). doi:10.1016/j.entcs.2006.12.027

Vanderhaegen, F., Polet, P., Zhang, Z., & Wieringa, P. A. (2002). Barrier removal study in a railway simulation, *PSAM 6,* Puerto Rico, USA.

Vasca, F., Camlibel, M. K., Iannelli, L., & Frasca, R. (2009). A new perspective for modeling power electronics converters: complementarity framework. *IEEE Transactions on Power Electronics*, *24*(2), 456–468. doi:10.1109/TPEL.2008.2007420

VDI. (2010). *4006 Part 3: Human reliability - Methods to analyse events regarding human behaviour, draft*. Düsseldorf, Germany: VDI-Gesellschaft Produkt- und Prozessgestaltung.

Vemuri, A. T., Polykarpou, M. M., & Diakourtis, S. A. (1998). Neural network based fault detection in robotic manipulators. *IEEE Transactions on Robotics and Automation*, *14*(2). doi:10.1109/70.681254

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., & Yin, K. (2003). A review of process fault detection and diagnosis, Part III: Process history based methods. *Computers & Chemical Engineering*, *27*, 327–346. doi:10.1016/S0098-1354(02)00162-X

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. N. (2003). A review of process fault detection and diagnosis, Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, *27*, 293–311. doi:10.1016/S0098-1354(02)00160-6

Verhallen, T., & van de Goor, A. (1992). Functional testing of modern microprocessors. In, *3rd European Conference on Design Automation* (pp. 350–354). Brussels, Belgium: IEEE Computer Society Publications.

Vernadat, F., Percebois, C., Farail, P., Vingerhoeds, R., Rossignol, A., Talpin, J., et al. (2006). The TOPCASED Project - A Toolkit in OPen-source for Critical Applications and System Development. In *Data Systems In Aerospace (DASIA)*. Berlin: European Space Agency (ESA Publications).

Vesley, W., Dugan, J., Fragola, J., Minarick, J., & Railsback, J. (2002). *Fault Tree Handbook with Aerospace Applications*. Washington, D.C.

Video and resources available on internet, www.youtube.com

Vinter, J., Bromander, L., Raistrick, P., & Edler, H. (2007). *FISCADE - A Fault Injection Tool for SCADE Models*. Paper presented at the 3rd Institution of Engineering and Technology Conference on Automotive Electronics.

Visser, W., Havelund, K., Brat, G., Park, S., & Lerda, F. (2003). Model Checking Programs. *International Journal on Automated Software Engineering*, *10*(2), 203–232. doi:10.1023/A:1022920129859

Vromans, M. J. C. M. (2005). *Reliability of Railway Systems*. The Netherlands: TRAIL Research School.

Wang, Z., Liu, Y., & Griffin, P. J. (1998). A combined ANN and expert system tool for transformer fault diagnosis. *IEEE Transactions on Power Delivery*, *13*(4), 1224–1229. doi:10.1109/61.714488

Weiss, D. M. (2008). The product line hall of fame. In *SPLC '08: Proceedings of the 2008 12th International Software Product Line Conference*, (p. 395), Washington, DC, USA: IEEE Computer Society.

Wen, C.H.-P. & Wang Li-C & Cheng K.-T. (2006). Simulation-based functional test generation for embedded processors. *IEEE Transactions on Computers*, *55*(11), 1335–1343. doi:10.1109/TC.2006.186

Williams, J. C. (1986). A proposed Method for Assessing and Reducing Human error. *Proceedings of the 9th Advance in Reliability Technology Symposium. pp. B3/R/1-B3/R/13*, University of Bradford

Winter, K. (2002). *Feasibility study on control table verification. SigTools-041, version 0.4*, October 2002.

Winter, K. (2002). *Model checking control tables: the ASM-NuSMV approach. SigTools.039, version 0.1*, October 2002.

Winter, K., & Robinson, N. J. (2003). Modelling large railway interlockings and model checkingsmall ones. In Oudshoorn, M. (Ed.), *Proc. of Australasian Computer Science Conference* (ACSC2003).

Winter, K., (2012) *Symbolic Model Checking for Interlocking Systems, Railway Safety, Reliability and Security: Technologies and Systems Engineering*,

Winter, K., Johnston, W., Robinson, P., Strooper, P., & van den Berg, L. (2005). Tool support for checking railway interlocking designs. In Cant, T. (Ed.), *Proc. of the 10th Australian Workshop on Safety Related Programmable Systems (SCS'05), volume 55*, (pp. 101-107). Australian Computer Society, Inc.

Winter, K. (2008). *Model Checking Abstract State Machines*. VDM Verlag.

Woodcock, J., Larsen, P. G., Bicarregui, J., & Fitzgerald, J. (2009). Formal methods: Practice and experience. In *ACM Computation Survey* (pp. 1-36).

Yamashita, Y., Mitsuru, I., Tatsuya, K., Arata, M., Daisuke, I., & Kazusaku, F. (2011) *Advanced active control of a contact force between a pantograph and a catenary for a high-speed train,* Procedings of the 9th World Congress of Railway Research, May 22-26 May 2011

Yen, G. G., & Lin, K. C. (2000). Wavelet packet feature extraction for vibration monitoring. *IEEE Transactions on Industrial Electronics*, *47*(3), 650–667. doi:10.1109/41.847906

York (2002): *Potential electromagnetic interferences to radio services from railways, final report for Radiocommunications Agency AY 4110*. York EMC Services LTD.

York (2004): *Improved methods for the measurement of radiofrequency emissions from railways AY 4365*. York EMC Services LTD.

Zadeh, L. A. (1968), Fuzzy algorithms, in *Information and Control*.

Zeng, D., Chawathe, S. S., Huang, H., & Wang, F. (2007), Protecting Transportation Infrastructure, in *IEEE Intelligent Systems, 22*, 8-11.

Zhou, F. B., Duta, M. D., & Henry, M. P. (2002). Remote condition monitoring for railway point machine. *Proceedings of the ASME/IEEE Joint Rail Conference*, (pp. 103-108).

Zimmermann, A., & Hommel, G. (2005). Towards modeling and evaluation of etcs real-time communication and operation. [Parallel and distributed real-time systems]. *Journal of Systems and Software*, *77*(1), 47–54. Available from http://www.sciencedirect.com/science/article/B6V0N-4D985FD-2/2/088a5957bc822c094e50eee203a4258fdoi:10.1016/j.jss.2003.12.039

zu Hörste, M., & Schnieder, E. (1999). Modelling and simulation of train control systems using Petri nets. In FMrail workshop (Vol. 3).

About the Contributors

Francesco Flammini got with honours his laurea (2003) and doctorate (2006) degrees in Computer Engineering from the University Federico II of Naples. Since October 2003, he has worked in Ansaldo STS (Finmeccanica) on the safety and security of rail-based transportation infrastructures. He has taught Computer Science and Software Engineering as an Adjunct Professor at the University of Naples as well as seminars on computer dependability and infrastructure security in post-degree courses. He has co-authored several books and more than 50 scientific papers published in international journals and conference proceedings. He has served as the chairman, a PC member and an editor for several international conferences and journals. He is a Senior Member of the IEEE, an ACM Distinguished Speaker, and the Vice-Chair of the IEEE Computer Society Italy Chapter. He is also member of: the European Workshop on Industrial Computer Systems Reliability, Safety and Security (EWICS TC7), FME (Formal Methods Europe) and ERCIM WG on Formal Methods for Industrial Critical Systems (FMICS).

* * *

Iñigo Adin is a researcher at CEIT. He received his MSc. Degree in Electronics Engineering in 2003 and his PhD in 2007 at the University of Navarra. From 2003 to 2007 he worked towards his PhD focused on CMOS RF front-ends for multistandard wireless applications in the 5GHz U-NII band. Other projects in the field of RFICs have been the design of the ESD protection for a low power front end for EPSON. He is presently engaged in the design of a safety critical receiver for a ERTMS BTM for high speed trains and he is coordinating the EC FP7 project TREND (contract number 285259) - Test of rolling stock compatibility for cross-domain interoperability. He is author or co-author of 1 patent, 2 technical books and 19 articles in journals and conferences. He is also an associate professor at TECNUN (Engineering School of Universidad de Navarra) lecturing circuit electronics topics.

Benedetto Allotta (S'90–M'92) was born in Agrigento, Italy, in 1963. He received the Laurea degree in mechanical engineering from the University of Pisa, Pisa, Italy, in 1987, and the Ph.D. degree in robotics from the Scuola Superiore Sant'Anna, Pisa, in 1992. From 1993 to 2001, he was an Assistant Pro-fessor of applied mechanics, □rst in the Advanced Robotics Technology and Systems (ARTS) Laboratory, and then in the Perceptual Robotics (PERCRO) Laboratory, Scuola Superiore Sant'Anna. From 1996 to 2000, he taught courses in the School of Engineering, University of Pisa. Since 1998, he has also been teaching courses at the University of Florence, Florence, Italy, where, since 2001, he has been an Associate Professor in the Section of Applied Mechanics, Department of Enegetics Sergio Stecco. He currently teaches courses in the area of automation and robotics. His current research interests include automation in transport systems, hardware-in-the-loop (HIL) simulation, control of robots, and mecha-

tronics. He is the author or co-author of more than 120 publications, including 30 papers published in international journals. He is the holder of two international patents. He is also responsible for several research grants and contracts from public agencies as well as private companies for a total amount of some hundred thousand Euros per year.

Silvio Baccari was born in 1975 in Benevento, Italy where currently he lives. He received the Master degree (Laurea') cum laude in Computer Engineering from the University of Sannio in Benevento, Italy. Since 2009, He is a Ph.D. student in Information Engineering at the University of Sannio, Benevento, Italy. His current research interests include hardware-in-the-loop systems, Real Time Model Predictive Control with applications to power electronics and FPGA and Microcontroller based platforms for rapid control prototyping of high efficiency AC/DC and DC/DC switching converters for high brightness LED. Mr. Baccari is a student member of the IEEE Control Systems Society.

Alfredo Benso received the MS degree in computer engineering and the PhD degree in information technologies, both from Politecnico di Torino, Italy, where he is working as a tenured associate professor of computer engineering. His research interests include DFT, BIST, and dependability. He is also actively involved in the Computer Society, where he has been a leading volunteer for several projects. He is a Computer Society Golden Core Member, and a senior member of the IEEE.

Jens Braband has obtained a doctorate degree in stochastic modeling from TU Braunschweig in 1992. He joined the Rail Automation business unit of Siemens AG as a safety expert and is currently Principal Expert for RAMSS. Since 1997 he is accredited as Independent Safety Assessor (ISA) by the German Federal Railway Office (EBA). In standardization he has contributed to may IEC and CENELEC standards. He currently represents the European Railway Manufacturers Association UNIFE in safety-related matters at the European Railway Agency (ERA). In 2004 he was awarded a honorary professorship at TU Braunschweig in the field of "Risk and Safety Analysis of Transportation Systems".

Giulio Cammeo was born in Foggia, Italy. He received the Laurea degree in Computer Science Engineering on July 2005 from the University of Sannio, Benevento, discussing a thesis on formation control and collision avoidance in mobile agent system. Since 2005 until 2008 he has worked with the Group for Research on Automatic Control Engineering at the Department of Engineering of the University of Sannio. His major activity was focused on the hardware in the loop systems for testing of the electronic control units, also collaborating with companies for testing traction control units using a realtime simulator based on multi-cpu and fpga architectures. Since 2008 he is with Ansaldobreda, Napoli, working on traction control unit software for induction motors and permanent magnetic synchronous motors.

Stefano Di Carlo received the MS degree in computer engineering and the PhD degree in information technologies from the Politecnico di Torino, Italy, where he has been an assistant professor in the Department of Control and Computer Engineering since 2008. His research interests include DFT, BIST, and dependability. He is a golden core member of the IEEE Computer Society and a member of the IEEE.

Alessandro Fantechi has studied Computer Science at University of Pisa in the late seventies, with a scolarship at the Scuola Normale Superiore di Pisa, earning a Laurea Degree in Computer Science

at the University, together with the Diploma of the Scuola Normale, in November 1978. He has then conducted research in the field of Software Engineering, concentrating in the last twenty years on the applications of formal specification and verification methods. His main current research interests are on industrial applications of model checking and on formal aspects of product line engineering. He has been affiliated with IEI - CNR in Pisa, University of Pisa, and, since 1995, University of Florence, where he is full professor and teaches courses on Embedded Systems, Software Dependability and Theoretical Computer Science to Engineering students. He maintains research collaborations with ISTI - CNR in Pisa, and Paristech Telecom in Paris. He has maintained strict relations with industries as well, starting from his one year and a half early experience in Olivetti in the early eighties, and then within several research, teaching and consulting collaborations with main Italian companies, such as Ansaldo, Alenia, Altran, General Electric Transportation Systems and Italian State Railways, as well as several small and medium enterprises. Most of these companies are active in the fields of safety critical computer systems, and A. Fantechi have had hence the opportunity to mature a significant experience, in particular on the Ada programmming language, on the industrial applications of Formal Methods, on Software Certification, and in the railway signalling domain. He has partecipated to European research projects, namely PAPS (Portable Ada Programming System - 1981-82), AdaFD (Ada Formal Definition 1985-1987), LOTOSPHERE (1989-92), GUARDS (Generic Upgradable Architecture for Dependable Systems - 1996-99), MODTRAIN-MODCONTROL (2004-2008), Sensoria (2005-2009). He is member of AICA, FME, IFIP WG 6.1, and has been coordinator of the ERCIM FMICS Working Group from 2008 to 2011.

Angela Di Febbraro was born in Ronco Scrivia, Italy, on November 23rd, 1963. She received the Laurea degree in Electronic Engineering in 1987, and the Ph.D. Degree in Computer Science and Electronic Engineering in 1992, both from at the University of Genoa. She has been Assistant Professor at the University of Genoa, and from 1998 to 2005, Associate Professor at the Polytechnic of Turin, Italy. Since 2005 she is Full Professor of Transportation at the University of Genoa. Member of EURO Working Group on Transportation since 1996, she has been co-editor of different special issues of international scientific journals, and reviewer of both international scientific books, and papers submitted to different journals and periodic international conferences, especially for IEEE and IFAC. Her main research interests are in modeling, optimization, and control of freeway, interurban, and urban transportation systems, and of logistic systems.

Christian Dufour received a Ph.D. degree from Laval University, Quebec, Canada in 2000. He joined Opal-RT Technologies in 1999 where he is the lead researcher in electric system simulation software. Before joining Opal-RT, he worked on the development of Hydro-Quebec's HYPERSIM real-time simulator, as well as MathWorks' SimPowerSystems blockset. His current research interests are related to algorithmic solutions for the real-time simulation of power systems and motor drives in RT-LAB, the real-time platform of Opal-RT Technologies.

Lars Ebrecht Dipl.-Inform. Lars Ebrecht is born in 1974. He studied electrical engineering and computer science at the Technical University of Braunschweig and graduated in 2002. From 1998 to 2000 he was employed by DAVID ltd. Since 2002 he is working as scientific staff at the German Aerospace Center in the Department of Railway Systems of the Institute of Transportation Systems in Braunschweig. His main research is focused on the development of innovative methods for testing of the new European Train Control System ETCS.

Johannes Feuser received the Dipl.-Ing. degree in electrical engineering from the University of Bremen, Bremen, Germany, in 2007. He is currently working toward the Ph.D. degree in computer sciences at the Research Group Operating Systems, Distributed systems of the University of Bremen. His research focus is the domain-specific modeling of safety-critical embedded real-time systems, especially train control systems. Before that he worked as scientist at Bremen University in the field of service robotics.

Matthias Güdemann studied computer science and mathematics at the University of Augsburg. He completed his diploma thesis on optimization of automatic palletizing algorithms at KUKA AG in 2005. From 2005 to 2009 he worked as a researcher at the chair of software engineering and programming languages at the University of Augsburg on the topics of organic computing and safety analysis. He is currently finishing his Ph.D. thesis on "Qualitative and Quantitative Model-Based Safety Analysis", which extends the existing approaches and allows for direct integration of quantitative aspects in formal models.

Axel Habermaier studied software engineering at the University of Augsburg, the Technical University of Munich, and Ludwig-Maximilians-University Munich. He completed his master's thesis on "The Model of Computation of CUDA and its Formal Semantics" in 2010. He is now a research assistant at the Institute for Software and Systems Engineering at the University of Augsburg, where he is working in the field of formal safety analysis techniques. Furthermore, he is continuing his research on the application of formal methods to the development of GPU-accelerated massively parallel programs.

Malte Hammerl is born in 1980. He studied engineer of transport and transportation systems at the Technical University of Dresden and graduated in 2006. From 2006 to 2011 he was working as scientific staff at the German Aerospace Center in the Department of Railway Systems of the Institute of Transportation Systems in Braunschweig. His main research focus is the evaluation of railway specific working environments and the assessment of human reliability. In 2011 he has got his phd in engineering for his phd-thesis about the analysis of human factors and reliability in railways.

Mark Hartong, PE Mark Hartong is a Senior Electronics Engineer in the Office of Safety of the Federal Railroad Administration (FRA), US Department of Transportation. The FRA is the regulatory and enforcement agency responsible for promoting safe and successful railroad transportation within the United States, and advancing the executive branch policies regarding freight and passenger rail. As an interdisciplinary electronics engineer, he serves as the agency's senior technical authority with respect to the application of safety and security critical electronics and software for use in the railroad environment. He received his doctorate in Information Technology in 2009 from George Mason University. He also has a MSc in Software Systems Engineering with a Certificate in Information Systems Security, a MSc in Computer Science, and a BSc in Mechanical Engineering from George Mason University, the US Naval Postgraduate School, and Iowa State University respectively. Mark is also a Registered Professional Engineer.

Anne Haxthausen is associate professor at DTU Informatics, Technical University of Denmark. She received the M.Sc.E. and Ph.D. degrees from DTU in 1985 and 1989, respectively. From 1988 to 1994 she worked at Dansk Datamatik Center and CRI A/S in Denmark, and then she returned to DTU. Haxthausen has more than 25 years of experience in theory and practice of formal methods for software

development. She was one of the main investigators of several formal softwaremethods in the ESPRIT funded projects RAISE 1985-90, LaCoS 1990-94, and CoFI 1995-2004. Industrial applications of her research focus on safety-critical applications, especially for railways. She participated in the ESPRIT FMERail project 1998-99, and currently she is one of the key persons in the RobustRailS project concerning Robustness in Railway Operations, funded 2012-15 by the Danish Council for Strategic Research. Furthermore, she is a consultant for Rail Net Denmark and the Danish Transport Authority.

Luigi Iannelli was born in Benevento, Italy, in 1975. He received the Master degree (Laurea) in computer engineering from the University of Sannio, Benevento, in 1999, and the Ph.D. degree in information engineering from the University of Napoli Federico II, Naples, Italy, in 2003. During 2002 and 2003 he visited, as a Guest Researcher, the Department of Signals, Sensors, and Systems, Royal Institute of Technology, Stockholm, Sweden. He was a Research Assistant at the Department of Computer and Systems Engineering, University of Napoli Federico II, and since 2004, he has been an Assistant Professor of automatic control with the Department of Engineering, University of Sannio, Benevento. His current research interests include analysis and control of switched and nonsmooth systems, and automotive control and applications of control theory to power electronics. Dr. Iannelli is a member of the IEEE and the SIAM.

Nina Jellentrup is born in 1983. She studied Psychology at the University of Oldenburg and graduated in 2008. From 2008 to 2009 she was working research projects in Industry. From 2009 to 2011 she was working as scientific staff at the German Aerospace Center in the Department of Railway Systems of the Institute of Transportation Systems in Braunschweig. Her main research focus is the field of Rail Human Factors especially the development of rail specific interactive systems with good usability. She was involved in several usability projects with railway operators and manufacturers.

Raffaele Malangone He graduated in electronic engineering in 2001 and attended a Master in Safety Critical System in 2004 . From 15/4/2002 he is working at RFI (Italian Railway Company) on Automatic Train Control project as system engineer on ETRMS and SCMT systems, in particular for the tests and the planning of manutenability, RAMS and for the formal languages specification. He has been member of the Italian Technical Commission for the putting in service of the High Speed Lines. He has also worked in 2001 in Aerospace Alenia Company. He is eternal member and professor at Salerno University for Transport Department and Author and co-author for many scientific international papers.

Jaizki Mendizabal is a lecturer at Tecnun (University of Navarra, Spain), and a researcher in the Electronics and Communications Department at CEIT (Spain). He received his MSc and PhD degrees in Electrical Engineering from Tecnun in 2000 and 2006 respectively. He joined Fraunhofer IIS-A (Germany) from 2000 to 2002 and SANYO Electric Ltd (Japan) from 2005 to 2006 as RF-IC designer. He obtained his PhD in the field of monolithic RF design for GNSS systems. He currently works in CEIT where his research interests include GNSS and safety-critical systems for the railway industry. He has participated in more than 8 research projects, has directed 2 doctoral theses, is author or co-author of 1 patent and 22 scientific and technical publications in national and international journals and conferences and is the author of the book “GPS and Galileo Dual RF Front-end receiver and Design, Fabrication, & Test”.

Juan Meléndez received MSc. and Ph.D. degrees from the Engineering School of University of Navarra (TECNUN), Spain, in 1998 and 2002 respectively. Since 1998 he has researched in the field of communication systems for several organizations such as Fraunhofer Institut fur Integrierte Shaltungen in Erlangen (Germany), Hitachi Semiconductors Europe Ltd. (United Kingdom), ABB Automation products (Spain), CAF (Spain), etc. Since 2006 he is in charge of the safety critical embedded system laboratory in CEIT. Currently he is also assistant professor of "Embedded systems for biomedical applications" and "Electromagnetic Compatibility" at TECNUN (University of Navarra). He is author and coauthor of two books, a patent and thirty articles in technical journals and international congresses.

Jon Mendizabal Samper joined the Electronics and Communications Department of CEIT in 2007, where he is currently a PhD candidate. He received his M.S in Electronics Engineering, majoring in Communications, in 2007 from the University of Navarra (Spain). His research, as well as technical interest, is focused on safety software system design and development methods and tools. At present, he is participating in an ERTMS (European Rail Traffic Management System) project. He is also a teacher of Protocols and Software Engineering lecture.

Michael Meyer zu Hörste is born in 1969. He studied mechanical engineering in at the Technical University of Braunschweig and graduated in 1995. From 1995 to 2001 he was working at scientific staff at the Institute of Control and Automation Engineering of the Technical University of Braunschweig in the field of railway safety. In 2004 he has got his phd in mechanical engineering for his phd-thesis about modelling and simulation of the generic behaviour of train control systems. Since 2001 he is working as scientific staff at the German Aerospace Center in the Department of Railway Systems of the Institute of Transportation Systems in Braunschweig. His main research focus is the European Train Control System (ETCS), the European Rail Traffic Management System (ERTMS), interlocking and train localisation.

Vincenzo Mungiguerra was born in Napoli, Italy in 1960. He got the Laurea degree in Electronic Engineering at the University of Napoli in 1986. Since 1986 he worked with the Development Department at Ansaldo Trasporti, now AnsaldoBreda. He worked on propulsion control for heavy vehicles (loco E402A, ETR500, Emu Norway), in mass transit vehicles (Metro Milano, Metro Napoli, Metro Roma, Metro Madrid), LRV (Copenhagen), tram vehicles (Oslo, Birmingham), and trolley bus (Genova, Napoli). He contributed to the development of a simulator for the entire train propulsion system (from pantograph to wheel-track), integrated with a board simulator in which the real control software is implemented. From 2007 to 2009 he operated as system integrator for the control software of several train components of Metro Madrid (train control, propulsion control, pneumatic brake control and diagnostic unit). Form 2009 to 2010 he worked on electrical traction systems for mass transit applications, contributing to win the tender of Miami, Copenhagen 2nd generation and Honolulu. Since 2010 he coordinates the team dedicated to the development of control algorithms for propulsion and auxiliary converters.

George Nikandros is an electrical engineer with over 33 years experience in the railway signalling industry. He was a foundation member of the Australian Computer Society's National Technical Committee on Safety-Critical Systems when it was established in 1992; a committee which evolved into the Australian Safety Critical Systems Association in 2002. He chaired that association from its inception in 2002 until June 2010. He is a member of the Railway Technical Society Australasia and member of

the Queensland chapter committee since its formation in 1998 and chaired that committee from 1999 to 2004. He is a Chartered Member of Engineers Australia, a Fellow of the Institution of Railway Signal Engineers, a Senior Member of the Australian Computer Society and member of the Risk Engineering Society. George has published papers and a co-author of the book "New Railway Environment – A multi-disciplinary business concept".

Frank Ortmeier studied mathematics and physics at the University of Augsburg. He received his diploma in 2001 and proceeded to work as a researcher at the chair of software engineering and programming languages at the University of Augsburg. He finished his Ph.D. thesis on "Formal Model-Based Safety Analysis" in 2005. From 2005 to 2009 he worked as post-doc at the University of Augsburg in various research projects, in particular organic computing, safety analysis, and robotics. Since september 2009 he is head of the chair of the Computer Science and Engineering group at the Otto-von-Guericke University of Magdeburg.

Federico Papa was born in Genoa, Italy on January 25th, 1985. In 2009, he received the Laurea Degree in Transportation Engineering and Logistics with a dissertation on the design and the development of an innovative tool for assessing the security of rail infrastructures. Since 2010 he has been at University of Genoa as a grant holder for a research on critical analysis of systems to mitigate the risks related to vandalism and terrorism attacks to railway assets, in collaboration with Ansaldo STS SpA. Currently, he is a Ph.D. student and is involved in European research projects aiming to improve railway safety and security.

Jan Peleska is professor for computer science (operating systems and distributed systems) at Bremen University in Germany. Before that he worked as Senior Software Designer and later on as department manager and consultant in the fields of fault-tolerant systems, distributed systems and database systems and, in particular, safety-critical embedded real-time systems. His habilitation thesis focusing on Formal Methods for the development of dependable systems was completed in 1995. He is co-founder of Verified Systems International GmbH, a company providing tools and services in the field of safety-critical system development, verification, validation and test. His research interests include formal methods for the development of dependable systems, automated model-based testing and V&V for safety-critical systems, with applications in the avionic, automotive and railway domains.

Mario Porzio, was born in Napoli, Italy, in 1959. He received the Laurea degree in eletronic engineering from the Università di Napoli Federico II, Napoli, Italy, in 1983. He has been with Ansaldo Trasporti spa (now Ansaldobreda spa), Naples, Italy, since 1984, where he is currently a Senior Engineer. His main experiences are on power converter control for locomotives, both ac/dc and dc/dc power converter control, and induction motor control. His research interests include modeling of traction electrical and mechanical systems.

Luca Pugi, born in 1974 in Florence, received the degree in mechanical engineering in 1999 from the University of Florence, Italy, and the Doctorate Degree in Applied Mechanics in 2003 from the University of Bologna, Italy. He is currently a Researcher in the Department of Energetics Sergio Stecco, University of Florence, Italy, where he is involved in design and simulation of mechatronic systems, mainly for vehicle applications collaborating with relevant industrial partners. Currently he teaches in courses of engineering concerning mechatronics and modelling of dynamical systems at University of

Florence. He is also a consultant and a system developer for fluid-controlled mechatronic systems with specific emphasis on pneumatic and hydraulic braking plants. As VIS for brake and on-board vehicle subsystems (Italian Acronym for independent safety inspector-assessor) he is currently cooperating with Italcertifer SPA. He is the author or co-author of about 90 publications and winner of two awards from CIFI, the Italian association of railway engineers, and one gained at WCRR2011 (World Congress for Railway Research). As Reviewer Dr.Pugi has cooperated with journals of various organizations, including IEEE and IAVSD (International Association for Vehicle System Dynamics).

Gabriella Reale was born in Campobasso, Italy, in 1983. She received the first level degree in Computer Engineering in 2006 and the second level degree in Automatic Control Engineering in 2008 from University of Sannio, Benevento, Italy. Since 2009 she has been an Assistant Researcher of Automatic Control with the Department of Engineering, University of Sannio, Benevento. Her research interests include automotive control, control of power electronic systems, rapid control prototyping and real-time hardware-in-the-loop techniques for railway applications.

Wolfgang Reif is professor for software engineering at the University of Augsburg. He is dean of the Faculty of Applied Computer Science, and director of the Institute for Software and Systems Engineering. His research interests are software and systems engineering, safety, reliability and security, organic computing, and software-driven mechatronics and robotics. Prof. Reif is involved in numerous research projects both in fundamental as well as application oriented research.

Clive Roberts is a Professor of Railway Systems at the University of Birmingham and Director for Railway Research for the Birmingham Centre for Railway Research and Education. Over the last 14 years he has developed a broad portfolio of research aimed at improving the performance of railway systems. He leads the University's contribution in a number of large EPSRC, European Commission and industry funded projects. He works extensively with the railway industry in Britain and overseas. He currently leads a team of 11 research fellows and 16 PhD students.

Neil Robinson is a Consultant and Director of RGB Assurance, providing engineering and management consultancy services in the area of safety-related and high integrity systems, and is an Adjunct Professor in the School of Information Technology and Electrical Engineering at The University of Queensland, Australia. Neil's experience consists of 18 years in the area of safety-critical systems, including senior management roles with responsibility for assurance of System Safety, Verification and Validation, Occupational Health and Safety, Environment and Quality. Neil's experience also includes system safety engineering, systems engineering, management and consultancy to clients in Rail, Defence and Oil and Gas and other industries. Neil is a Member of the British Computer Society and Chartered Engineer.

Nicola Sacco was born in Borgosesia, Italy, on December 4th, 1976. He received the Laurea Degree in Electronic Engineering in 2000, and the Ph.D. Degree in Automatics and Computer Sciences for Transportation Systems in 2004, both from the Polytechnic of Turin, Italy. From 2004 to 2006, he held an Assistant Researcher position at Polytechnic of Turin and, and since 2006 he is Assistant Researcher at University of Genoa. His main research interests include theory, safety, and security of transportation systems, with particular attention to the performance and sensitivity analyses of car-sharing services,

urban traffic, and freight transportation. The results of such research activity have been published into about 40 papers, both as conference proceedings and journal papers.

Alessandro Savino received the MS degree in computer engineering and the PhD degree in information technologies from the Politecnico di Torino, Italy, where he has been a postdoc in the Department of Control and Computer Engineering since 2009. His main research topics are microprocessor test and software-based self-test.

Fabio Senesi He graduated in electronic engineering and has a PhD in Applied Electromagnetism. He work for RFI since 1995 and is a specialist in Developing Train Command and Control Systems, Safety Assessment and Signalling Railway design and Requirements Specification. Actually he is responsible of the Automatic Train Control Project for the Italian Signalling Systems. Previously he has also been Coordinator of sector Inspectorate and Control at ANSF Vice Coordinator of Sector Technical Standards at ANSF (National Safety Agency for Railway in Italy). He is an active member of European Railway Agency and of the Italian Railway Safety Agency. He has followed as responsible respectively of the Commission for the acceptance of Electronic Interlocking of Roma Termini Station (1996-1999) and ETCS Level 2 for Italian High Speed Lines (2002-2009). He has been Vice-president of the Commission for the Safety Acceptance of the all High Speed and responsible for development and homologation of many systems/products.

Gerhard Schellhorn studied Computer Science at the University of Karlsruhe. He got his PhD from the University of Ulm in 1999 on the topic of "Verification of Abstract State Machines". Since 2000 he is working as a senior researcher at the Institute for Software and Systems Engineering at the University of Augsburg, where he leads the formal methods group. His main interests are software engineering, formal specification and verification of software systems as well as safety and security analysis.

Joseph Silmon studied a Masters in Electronic and Electrical Engineering at the University of Birmingham. During this course he spent 13 months on exchange at the University of New South Wales, Sydney, Australia. Joe's specialisations at undergraduate level were primarily in power electronics and energy systems. Joe graduated in 2004 with a II(i) and started work in August of that year with the Mainline & Metros division of Bombardier Transportation, at Derby Carriage Works.After 1 year as a graduate engineer Joe chose to return to university to take up a research studentship, working on a Ph.D. thesis entitled "Operational industrial fault detection and diagnosis: railway actuator case studies". This work aimed to develop previous research on fault detection for low-cost, high-population assets such as railway switches, train doors and level crossing barriers, with intuitive methods which could aid maintenance staff in targeting their work in time to avoid costly in-service failures. The thesis was successfully defended in October 2009 and Joe graduated with a Ph.D. on 11th December of the same year. During 2008, Joe worked on the European Commission projects SELCAT and Innotrack in parallel with the final touches of his Ph.D. He became a contracted member of University staff in March 2009, funded by the InfraGuidER project which aims to produce environmental guidance for railway infrastructure managers. Joe's role in all of these projects has been as a systems modeller and analyst.

Paul Strooper is the Head of School and a Professor in the School of Information Technology and Electrical Engineering at The University of Queensland. He received the BMath and MMath degrees in Computer Science from the University of Waterloo, and the PhD degree in Computer Science in 1990 from the University of Victoria. His main research interest is Software Engineering, especially software verification and testing, and model-based approaches to software development and verification. He has had substantial interaction with industry through collaborative research projects, training and consultation in these areas. He was one of the General Chairs for the 2010 Asia-Pacific Software Engineering Conference (APSEC) and the 2009 Australian Software Engineering Conferences (ASWEC), the program chair for APSEC in 2002 and ASWEC in 2004 and 2005. He is member of Steering Committees for ASWEC and APSEC, and a member of the editorial board of the IEEE Transactions on Software Engineering and the Journal of Software Testing, Verification and Reliability.

Jörn Guy Süß received the Master of Computer Science from the Technical University Berlin in 1999 and worked as a software architect for several years before changing to academia. He is a research fellow at the University of Queensland. His research is focussed on model driven and generative approaches to software engineering, with a particular focus on the application of model constraint and transformation languages.

Markus Talg is born in 1982. He studied mathematics at the Technical University of Braunschweig and graduated in 2009. Since 2009 he is working as scientific staff at the German Aerospace Center in the Department of Railway Systems of the Institute of Transportation Systems in Braunschweig. His main research is focused on the development of innovative methods for functional risk assessment, the analysis of the human influence on railway safety, authorisation issues and safety standards like CENELEC and the Common Safety Methods on risk evaluation and assessment.

Francesco Vasca was born in Giugliano, Italy, in 1967. In 1995 he received the Ph.D. degree in Automatic Control from the University of Napoli Federico II. Since 2000 he is Associate Professor of Automatic Control at the University of Sannio, Benevento, Italy. His research interests include: analysis and control of switched systems (averaging, complementarity, dithering, real time hardware in the loop) with applications to power electronics; automotive control for transmissions and hybrid electric vehicles; simulation of manufacturing systems; formation control of multiagent systems. Since January 2008 he serves as Associate Editor of the IEEE Transactions on Control Systems Technology. Since 1994 he is a member of IEEE Control System Society and IEEE Power Electronics Society.

Almir Villaro is a researcher in the Electronics and Communications Department at CEIT and a teaching assistant at TECNUN (University of Navarra). He received his M.S in Electronics Engineering, majoring in Communications, in 2007 from the University of Navarra (Spain). After developing security and localization on Wireless Networks at Telecom y Novatecno S.A., his research is presently focused on the development of Test strategies for Dependable Systems and Fault Injection for the Validation of Fault Tolerant Systems. He has participated in research projects in close collaboration with industrial partners and public authorities, and has published seven technical papers in international journals and conferences.

Duminda Wijesekera is an associate professor in the Department of Information and Software Engineering at George Mason University, Fairfax, Virginia. During various times, his research interests have been in security, multimedia, networks, secure signaling (telecom, railway and SCADA), avionics, missile systems, web and theoretical computer science. He holds courtesy appointments at the Center for Secure Information Systems (CSIS) and the Center for Command, Control and Coordination (C4I) at George Mason University, and the Potomac Institute of Policy Studies in Arlington, VA. Prior to GMU he was at Honeywell Military Avionics, Army High Performance Research Center at the University of Minnesota, and the University of Wisconsin. His doctorates are in Computer Science and Logic from the University of Minnesota and Cornell University in 1997 and 1990 respectively.

Kirsten Winter received the Ph.D. degree in computer science from the Technical University Berlin, Germany, in 2001. She currently holds a position as a research fellow at the University of Queensland, Australia. Her research interests includes the verification of software and hardware systems, mainly model checking, as well as formal modeling notations.

Christian Wullems is a currently a postdoctoral researcher at the Cooperative Research Centre for Rail Innovation, Australia. He is currently leading a number of cooperative industry projects in the area of rail safety. He received his Ph.D. from the Information Security Research Centre at Queensland University of Technology, where he investigated the security of location acquisition systems including GNSS, and their use in safety and financially critical applications. Prior to his present appointment, Christian Wullems was the technical director of Qascom S.r.l., Italy and was responsible for the preparation and technical management of a number of European FP7 research projects focused on the development of anti-spoofing techniques for civilian GNSS receivers. His main research interests include the application of cryptography to location determination systems for authentication and access control, civilian anti-spoofing techniques for GNSS, communications security and GNSS in the context of rail / road safety.

Index

A

Accident Sequence Evaluation Program (ASEP) 61
Advanced Civil Speed Enforcement System (ACSES) 5-6
Advanced Railroad Electronics System (ARES) 4
Advanced Train Control Systems (ATCS) 4
Analogue to Digital Converter (ADC) 168
Ansaldo STS 175-176, 197, 206, 217
Application Logic 177
Application Program Interface (API) 29, 45
Arcing 264
Artificial Neural Network (ANN) 346
Association of American Railroads (AAR) 4, 19-20
Automatic Programming 185
Automatic Test Generation (ATG) 286
Automatic Test Pattern Generator (ATPG) 205
Automatic Train Control (ATC) 267, 291, 385
Automatic Train Protection (ATP) 267
Availability 402

B

Balise Transmission Module (BTM) 128, 170
Barrier 329
Barriers 59
Best Practices 206
Binary Decision Diagrams (BDDs) 303
Binary Frequency Shift Keying (BFSK) 170
Binary Phase Shift Keying (BPSK) 169
Bit Error Rate (BER) 153, 167-168
Burlington Northern (BN) 4

C

Catenary 225
Code of Federal Regulations (CFR) 21
Collision Avoidance System (CAS) 6
Commercial-Off-The-Shelf (COTS) 200
Common Safety Methods (CSMs) 54, 64
Communications-Based Train Management (CBTM) System 5
Compiler Validation 42
Computation Tree Logic (CTL) 287
Computer Aided Design (CAD) 192
Computer based Train Management (CBTM) System 6
Conformity 117
Contextual Inquiry 321
Continuous Phase Binary Frequency Shift Keying (CPFSK) 169-170
Control Software 222
Control Table 184
control vector (CV) 212
Corridor 5-6, 119, 127
Covert Channels 31

D

Deductive Cause Consequence Analysis (DCCA) 99
Discrete Event Systems (DESs) 360
Domain Specific Language (DSL) 25
Driver Machine Interface (DMI) 36, 117
Driving Support System - See Sistema Supporto Condotta (SSC).

E

Electromagnetic Compatibility (EMC) 161
Electromagnetic (EM) 153
Electromagnetic Fields (EMF) 160
Electromagnetic Interferences (EMI) 151, 156
Electro-mechanical Switch Actuator 350
Electronic Control Unit (ECU) 222, 234
Electronic Train Management System (ETMS) 5-6
Error Context Data (ECD) 212
Error of Commission 331-332, 334, 339
European Committee for Electrotechnical Standardization (CENELEC) 199, 283
European Railway Agency (ERA) 2, 54, 121

European Railway Traffic Management System (ERTMS) 157
European Train Control System (ETCS) 2, 116-117, 383, 392
European Vital Computer (EVC) 395
EVC Application Executive (APEX) 29
Expert Evaluation 323
Extended System Model 85-86, 89, 93-95, 97, 99-100, 114

F

Failure Mode and Effect Analysis (FMEA) 129
Failure Modes, Effects and Criticality Analyses (FMECA) 55
Fault Coverage Computation 216
Fault Diagnosis 342
Fault Injection 140
Fault Models 203
Fault Simulation 350
Fault Tree Analysis (FTA) 56, 93
Federal Railroad Administration (FRA) 1, 19-21
Field Programmable Gate Array (FPGA) Boards 223
Finite Markov Chains 395
Fixed Causality Solver 236
Formal Fault Tree Analysis (FFTA) 93, 101
Formal Methods 283
Formal Methods of Software Engineering 114
Formal System Model 74
Form Fit Functional Interface (FFFIS) 130
Function 58, 64
Functional Testing 204
Fuzzy Logic 360, 365

G

General Electric Transportation Systems (GETS) 285
Ground Transportation Infrastructure (TI) 363

H

Hardware-in-the-Loop (HIL) 222
Hazard 99, 404
Hazard Rate (HR) 385
History-Based Diagnosis Method 346
Human-Barrier Interaction 329
Human Error 328
Human Factors 329
Human-Machine Interaction 339, 342
Human Reliability 60
Human Reliability Assessment (HRA) 328

I

Incremental Train Control System (ITCS) 5-6
Independent Test Lab 116, 127
Induction Motor 230
Industrial Condition Monitoring 341-342
Information Technology (IT) 156
Interactive Prototype 322
Interlocking 292, 298, 395
Interlocking Controllers 175
Interstate Commerce Commission (ICC) 9, 21
I/O-Safe Transition System (IOTS) 42

J

Juridical Recording Unit (JRU) 117

K

Kripke Structure 75

L

Level Crossing with Active Warning 422
Level Crossing with Passive Controls 400, 406-407, 411-412, 422
Life Cycle 385
Loop Transmission Module (LTM) 169
Losses Assessment (LA) 363
Low-Cost Level Crossing Warning Devices (LCLC-WDs) 399-401

M

Main Line Track Exclusion Addendum (MTEA) 14
Matlab Automotive Advisory Board (MAAB) 286
Meta-Metamodel 25-26
Microlok II 175-178, 184, 188, 192, 197
Microlok Interlocking Simulator System (MISS) 193
Microprocessor 199
Minimal Cut Sets 94, 96
Mitigation System Analysis (MSA) 363
Model Based Development (MBD) 285
Model Checking 70, 287, 303
Model-Driven Engineering (MDE) 25, 183
Multiple Input Signature Register (MISR) 215

N

National Railroad Passenger Corporation (Amtrak) 2

National Transportation Safety Board (NTSB) 3, 9, 20-21
North American Joint Positive Train Control (NAJPTC) System 6
NuSMV 68, 71, 74, 76-77, 79-80, 82, 84, 89-90, 95, 107-108, 111, 288-289, 296, 299-301, 303-305, 309-313, 315

O

Object Code Verification 41
Odometer 147
Office of Management and Budget (OMB) 10, 21
On-Board Unit (OBU) 116-117
openETCS 22-23, 25, 27, 29-32, 39, 41-42, 46-49
Operational Interoperability 119

P

Pantograph 225, 256
Paper Prototype 321
Partitioning 44
Passenger Rail Systems 358
Performance Shaping Factors 328
Phase Disconnection 230
Physical Interface 124
Pick-Up Model 231
Platform Independent Model (PIM) 25
Platform Specific Model (PSM) 25
Poison by Inhalation (PIH) -See Toxic by Inhalation.
Power Electronics Converters 248
Power Electronic Supply System 226
PTC Development Plan (PTCDP) 13
PTC Implementation Plan (PTCIP) 11
PTC Safety Plan (PTCSP) 13
Public Policy 9

Q

Qualitative-Based Diagnosis Method 345
Qualitative Trend Analysis (QTA) 347
Quantitative-Based Diagnosis Method 344
Quantitative Risk Analysis (QRA) 356
Queensland Rail (QR) 298-299

R

Radio Block Center (RBC) 117
Railroad Safety Advisory Committee (RSAC) 9, 20
Rail Safety Act 2006 (RSA) 403
Railway Association of Canada (RAC) 4
Railway Infrastructure Discretization 361
Railway Security Threats 357
Railway Signaling 291, 300
Railway Traction Control 221, 246
Real-Time Simulation 236
Real-Time Systems 113, 196, 220
Regression Testing 221-223
Right-side Failure 402
Risk 55
Risk Acceptance Criterion for Technical Systems (RAC-TS) 54
Risk Assessment 360
Risk Assessment Models 360
Risk Priority Numbers (RPN) 55
Risk Representation (RR) 371
Risk Score Matrix (RSM) 56

S

Saboteur 144
Safety 327
Safety Assessment 140
Safety Critical Application 220
Safety Critical System 298
Safety Integrity 401, 423
Safety Integrity Level (SIL) 59, 132, 401
Safety Optimization 104
Safety Principles 300
Safety-Related Application Rules (SAR) 56
Semi-Quantitative 56
Serviceability 116-117, 120, 127, 200
Signal Engineering 181
Signaling Systems 359
Signal to Noise ratio (SNR) 153, 167
Simulation 179
So Far As Is Reasonably Practicable (SFAIRP) 403, 417, 423
Software-Based Self-Test 206
Software Model Checking 289
Software Verification 214
Southern California Regional Rail Authority (SCRRA) 3
Surface Transportation Board (STB) 1, 21
Symbolic Model Checking 303
Synchronous Composition 77
System-in-the-Loop 249
System Specification 69

T

Technical Interoperability 117
Technical Interoperability Standards (TIS) 2

Technical System 55
Temporal Logics 89
Test Sequence Debugger (TSD) 117
Threat Analysis (TA) 364
Toxic by Inhalation (TIH) 11, 21
Track Layout 192
Traction Control Unit (TCU) 224, 234
Train Interface Unit (TIU) 117
Train Movement Control System - See Sistema Controllo Marcia Treno (SCMT).
Train Sentinel (TS) System 6
Transition Ordering 310
Triple Modular Redundancy (TMR) 140
Type Approval (TA) 15

U

Unified Modeling Language (UML) 106
Usability 323
Usability Test 323
User Interface 318
User Requirements 320

V

Validation 121, 144, 222
Variable Ordering 308
Verification 121, 241
Verification and Validation (V&V) Artifacts 23
Vital Train Management System (VETMS) 6
Vulnerability Analysis and Risk Assessment (VARA) 363

W

Wide Spectrum Formalism 52
Wind Tunnel Test 266
Worst-Case Execution Time (WCET) 40
Wrong-side Failure 411